

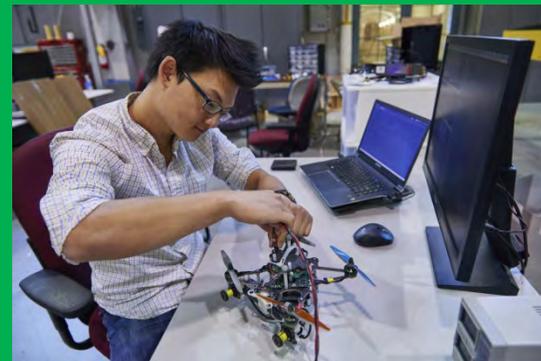
VOLUME 3

FALL 2015

ROBOTICS INSTITUTE

Summer Scholars (RISS) Working Papers

JOURNAL



Robotics Institute Summer Scholars Working Papers Journal

Volume 3 Fall 2015

Founding Editors

J. Andrew Bagnell
Reid Simmons
Rachel Burcin

Managing Editor

Rachel Burcin
rachel@cmu.edu

Assistant Managing Editors

Tess Hellebreker
Michael Lee
Cormac O'Meadhra

Cover Design & Document Layout

Alexandra Yau



We gratefully acknowledge the support of the National Science Foundation through the Research Experience for Undergraduates (REU) program (Grant # CNS 1263266).

Through the Summer Scholars Program, the Robotics Institute hosts the NSF Robotics & Intelligent Systems REU Site.

The Robotics Institute Summer Scholars Working Papers Journal is an annual publication of the Robotics Institute's Summer Scholars Program at Carnegie Mellon University. Copyright © Carnegie Mellon University 2015

Table of Contents

Summer Scholar Program	1
RISS Participating Labs 2015.....	2
Congratulations Cohort of 2015!	3
2015 NSF REU RISS Scholarship Recipients!	4
Thank You RISS Working Papers Journal Team	5
To Future Summer Scholars.....	6
About the Robotics Institute Summer Scholars Program	8
RISS 2015 Gallery	12
Working Papers.....	16
1. Sufyan Abbasi and Randy Sargent.....	20
<i>Visualizing Air Quality: The Environmental Sensor Data Repository Explorer</i>	
2. Gabriela Amaral Araujo de Oliveira, Tony Dear, and Howie Choset.....	24
<i>Development of an Online Learning Platform for an Undergraduate Robotics Course</i>	
3. Elias Bitencourt, Robert Paolini, and Matthew T. Mason.....	31
<i>3D Vision with Kinect and Similar Sensors</i>	
4. Laura Brooks, Heather Knight, and Reid Simmons	35
<i>Temporal Gestures for Expressive Motion</i>	
5. Nai Chen Chang, Xinlei Chen, and Abhinav Gupta.....	42
<i>Differentiating Singularity and Multiplicity in Web Images</i>	

6. Felipe G. Galiza and David S. Touretzky.....	46
<i>Analytical Inverse Kinematics of a 6-DOF Mantis Robot Arm and Modeling for Minimum Joints Load</i>	
7. Jonathan Y. Joo, Christopher Cunningham, Heather L. Jones, and William L. Whittaker	51
<i>Waypoint Sequencing for Planetary Rovers with Significant Time-Dependent Energy Constraints</i>	
8. Ariana Keeling, Robert Paolini, and Matthew T. Mason	59
<i>Autonomous Object Recovery in Manipulation Experiments</i>	
9. Arbaaz Khan, Shreyansh Daftry, and Martial Hebert.....	65
<i>Multi Modal Pose Fusion for Monocular Flight</i>	
10. Ratnesh Madaan, Robert Paolini, Erol Sahin, and Matthew T. Mason	69
<i>Predicting Orientations Under Manipulation Actions</i>	
11. Chirag Nagpal, Kyle Miller, Benedikt Boecking, and Artur Dubrawski	76
<i>An Entity Resolution Approach to Isolate Instances of Human Trafficking Online</i>	
12. Ravenna Thielstrom, Heather Knight, and Reid Simmons.....	82
<i>Generating Spatial Paths to Express Attentional Attitudes</i>	
13. Hadi Salman, Tony Dear, and Howie Choset.....	87
<i>A Novel Skidding Model for a Snakeboard</i>	
14. Hadi Salman, Chaohui Gong, and Howie Choset.....	94
<i>Shape Based Geometric Motion Planning for an Underactuated Highly-Articulated System</i>	
15. Minghan Wei.....	101
<i>Detecting Vehicles Based On Smartphone-collected Images</i>	
16. Xingchen Yu.....	107
<i>Methods and Experiments of Pedestrian Detection</i>	

Table of Illustrations



Cover Photo Credits:

1 - Courtesy of Sandeepa Veeramachaneni

2 - Courtesy of Debra Tobin

Photos used in “To Future Summer Scholars”:

<http://www.ri.cmu.edu>

Courtesy of Sandeepa Veeramachaneni

Courtesy of Debra Tobin

Photos used in “Robotics Institute Summer Scholars Program”

<http://www.tartanracing.org>

http://www.ri.cmu.edu/ri_static_content.html?menu_id=464

Photos used in “Photo Gallery” and in “Articles”

Courtesy of Debra Tobin

Carnegie Mellon THE ROBOTICS INSTITUTE

SUMMER SCHOLAR PROGRAM

At the core of the program are incredibly talented and dedicated faculty, graduate students, staff, and RISS alumni.

We are incredibly thankful for their support, participation, leadership, and vision that make this one of the best research experiences in robotics and intelligent systems in the world.



Thank you to the RISS 2015 Participating Labs

Auton Lab



Biorobotics Lab



Computer Vision Lab



CREATE Lab



Elastic Group

Field Robotics Center (FRC)



Illumination and Imaging Lab (ILIM)



Intelligent Coordination and Logistics Laboratory (ICLL)



Learning, Artificial Intelligence, and Robotics Laboratory (LAIRLab)



Manipulation Lab



NavLab



Personal Robotics Lab



Platypus, LLC.



Reliable Autonomous Systems Lab (RASL)

Robotics Academy



Robust Adaptive Systems Laboratory

Soft Robotics and Bionics Lab



Tekkotsu Lab



Congratulations Cohort of 2015!

Sufyan Abbasi	Vassar College (USA)
Gabriela Araujo	Federal University of Lavras (Brazil)
Clara Belitz	Bowdoin College (USA)
Elias Bitencourt	Maranhão State University (Brazil)
Laura Brooks	Tufts University (USA)
Nai Chen (Nadine) Chang	Columbia University (USA)
Arjav Desai	Delhi Technological University (India)
Felipe Gonçalves	The University Center of FEI (Brazil)
Abhinav Goyal	Johns Hopkins University (USA)
Tess Hellebreker	University of Texas, Austin (USA)
Jonathan Joo	California Institute of Technology (USA)
Ariana Keeling	Georgia Institute of Technology (USA)
Arbaaz Khan	Manipal Institute of Technology (India)
Evan Klei	Rochester Institute of Technology (USA)
Timothy Krentz	Valparaiso University (USA)
Michael Lee	Princeton University (USA)
Kevin Low	Carnegie Mellon University (USA)
Ratnesh Madaan	Indian Institute of Technology, Roorkee (India)
Cristina Morales	University of Puerto Rico (Puerto Rico)
Chirag Nagpal	Army Institute of Technology (India)
Cormac O'Meadhra	University College Dublin (Ireland)
Hadi Salman	American University of Beirut (Lebanon)
Arpit Saxena	Indian Institute of Technology, Kharagpur (India)
Griffin Tabor	Worcester Polytechnic Institute (USA)
Ravenna Thielstrom	Swarthmore College (USA)
Shantanu Vyas	University of Pittsburgh (USA)
Tairui Wang	Jiangsu University of Science and Technology (China)
Minghan Wei	Nanjing University of Science and Technology (China)
Yuzhang Wu	Nanjing University of Science and Technology (China)
Xingchen Yu	Nanjing University of Science and Technology (China)
Sam Zeng	Carnegie Mellon University (USA)
Zhong Zhe	Nanjing University of Science and Technology (China)
Siwei Zhu	Nanjing University of Science and Technology (China)

2015 NSF Research Experiences for Undergraduates RISS Scholarship Recipients



Laura Brooks
Tufts University



Nai Chen (Nadine) Chang
Columbia University



Tess Hellebrekers
University of Texas, Austin



Ariana Keeling
Georgia Institute of Technology



Evan Klei
Rochester Institute of Technology



Timothy Krentz
Valparaiso University



Kevin Low
Carnegie Mellon University



Cristina Morales
University of Puerto Rico



Ravenna Thielstrom
Swarthmore College

Thank You!

Special Thanks to the RISS Working Papers Journal Team



Sufyan Abbasi
Vassar College (USA)



Timothy Krentz
Valparaiso University (USA)



Gabriela Amaral Araujo de Oliveira
Federal University of Lavras
(Brazil)



Michael Lee
Princeton University (USA)



Clara Belitz
Bowdoin College (USA)



Kevin Low
Carnegie Mellon University
(USA)



Arjav Desai
Delhi Technological University
(India)



Chirag Nagpal
Army Institute of Technology
(India)



Tess Hellebrekers
University of Texas, Austin (USA)



Cormac O'Meadhra
University College Dublin
(Ireland)



Arbaaz Khan
Manipal Institute of Technology
(India)



Minghan Wei
Nanjing University of Science
and Technology (China)



Evan Klei
Rochester Institute of Technology
(USA)

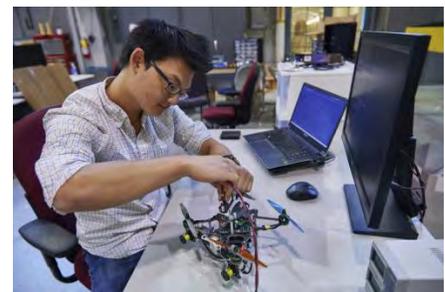
To Future Summer Scholars

To prospective and accepted scholars alike, we welcome you on your journey towards the Robotics Institute Summer Scholars (RISS) program. RISS is a phenomenal platform through which students can accelerate their undergraduate research experience in robotics. For many of the summer scholars, this was their first experience in a facility like the Robotics Institute.



Throughout the summer you will be surrounded by some of the world's leading roboticists, innovative research, and invaluable guidance and support. At the end of the summer, you will know much more than just how to conduct research. You will be a step ahead of your peers, having seen first-hand what cutting edge robotics requires.

The RISS program focuses on your individual research, which you will pursue under the guidance of your mentor and their graduate students. For many of you, this will be your first contribution to an active research project and first opportunity to make a tangible impact on the robotics community.



Research experience helps your graduate school application stand out so work diligently to make the most of your time here. To that end, you should strive to make substantial progress early on. Take it upon yourself to define milestones towards a final paper. Ultimately, the paper is a result of careful planning and foresight.

Your individual research project is not the only element of RISS, however. Throughout the summer you will be immersed in a community of exceptional students. This community will be nurtured through more formal events ranging from lab visits, guest speakers and graduate school application seminars to less formal cookie hours, frisbee games, and wakeboarding on the Allegheny River.



In RISS, the social aspects come hand in hand with your research. This diverse community of scholars from all over the world will nurture creativity and will help you explore a world beyond Carnegie Mellon University. You will share your experiences, your struggles, and your successes with the other RISS scholars. The summer flies by quickly, and you will be sad to part with your new friends.



Being one of the few REUs that accept international students, the RISS program is highly competitive. However, we encourage everyone to apply! You can't be accepted without sending in an application. Here is one small piece of advice – familiarize yourself with labs that interest and their research directions. When you put in this effort, it really shows genuine interest that the review committee is looking for. We wish you the best of luck!

In closing we say this to you: enjoy your time in CMU, meet as many people as you can, go to as many events as you can, see as much of Pittsburgh as possible; fully immerse yourself in the summer and you will not be disappointed. RISS provides the opportunity, but you must take the first step.

Sincerely,

Handwritten signature of Tess Hellebrekers in black ink.

Tess Hellebrekers

Handwritten signature of Mike Lee in black ink.

Mike Lee

Handwritten signature of Cormac O'Meadhra in black ink.

Cormac O'Meadhra

RISS Scholars 2015

Robotics Institute Summer Scholars Program

Carnegie Mellon's Robotics Institute Summer Scholars (RISS) program (<http://riss.ri.cmu.edu/>) is an eleven-week summer undergraduate research program that immerses a diverse cohort of scholars in cutting-edge robotics projects that drive innovative and have real-world impact. Launched in 2006, RISS is among the best and most comprehensive robotics research programs for undergraduates in the world. The quality and breadth of research, high-level of institute and university engagement, extensive professional development curriculum, graduate school application counseling, and alumni network create transformative experiences and remarkable post-program trajectories.



The RI Summer Scholars Program:

- 1) Immerses a highly motivated and diverse cohort of students (hereafter referred to as “scholars”) in a guided research process;
- 2) Challenges scholars to build an understanding of research philosophy that serves as a basis for creative problem-solving that transcends the summer research experience;
- 3) Introduces scholars to the highly interdisciplinary nature of robotics and the vast potential to impact and improve the world’s quality of life;
- 4) Provides professional development components that prepare students for successful transitions to graduate school and research careers;

- 5) Engages scholars in reflective service learning experience that imparts the value of broadening participation and engagement in STEM fields;
- 6) Inspires scholars to pursue careers in robotics and related fields, such as graduate programs, and equips them with new skills and knowledge; and
- 7) Helps scholars to build collaboration and lifelong connections with a dynamic global community of robotics researchers and entrepreneurs.

The Robotics Institute at Carnegie Mellon University is the largest university-affiliated robotics research group in the world. It offers a remarkably diverse breadth of research with an extensive range of applications. With hundreds of active research projects, together with both graduate and undergraduate programs, the Institute is a global leader in robotics research, education, and innovation. The Institute has the nationally recognized research capacity, educational programming, and student development experience to provide, through the RISS program, high-quality research experiences and a developmentally appropriate professional development curriculum to a diverse cohort of undergraduate students.



RISS Core Research Areas:

- 1) **Intelligence:** core AI technologies, motion planning, control theory, planning under uncertainty, POMDPS, game theory, data mining, and machine learning
- 2) **Perception:** computer vision, stereo processing, understanding lidar and 3D sensing, state-estimation, and pattern recognition
- 3) **Action:** work mechanisms, actuators, their design and control

Recent sample scholar projects have included:

- 3D Manufacturing of a Liquid Metal Microchannel for Soft Sensing and Actuation
- Auto-calibration and Hybrid Force/Position Control for the Cerberus Cardiac Robot
- Autonomous Object Recovery in Manipulation Experiments
- Design and Characterization of Map Based Lunar Rover Localization
- Generating Spatial Paths to Express Attentional Attitudes
- Highly Flexible and Stretchable Sensor Using Soft Optical Waveguides
- Improving Power and Vision Systems on Autonomous Quadrotors
- Monocular Visual Features for Fast Flight Through Forests
- New Visual Programmer Converter that Allows the Hummingbird Duo to Run Untethered
- Persistent Deployment of Micro-Aerial Vehicles
- Pothole Detection with Cell Phone Data
- Trajectory Following in GPS Denied Environments for UAVs
- Using Receding Horizon Control
- Visual Programmer and New Efficient File System

The RISS program also has a remarkable mentor and alumni community with wide participation and support from across the university. In 2015, over 35 researchers and professionals contributed to the RISS curriculum - presenting workshops, laboratory tours, and other programming elements. Over 50 members of the Robotics Institute (RI) community



participated as research mentors in the same period. Unique partnerships with robotics,

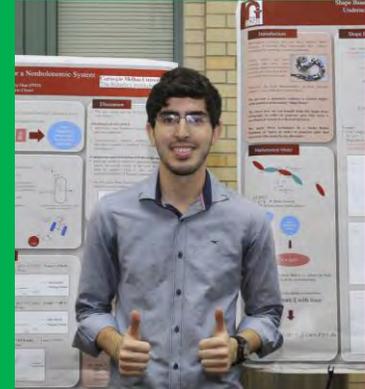
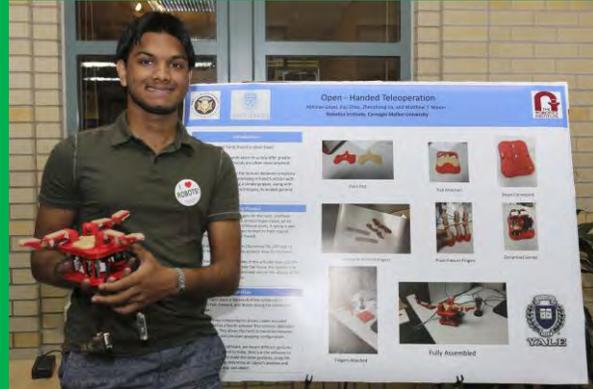
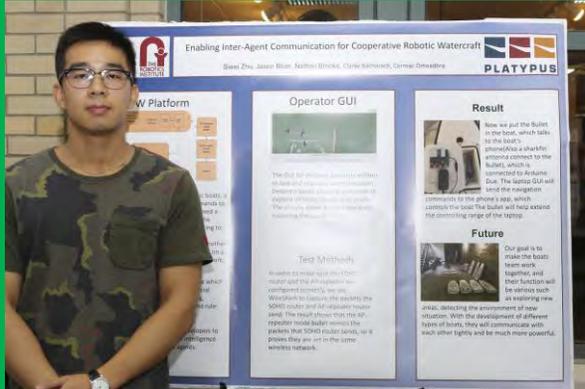
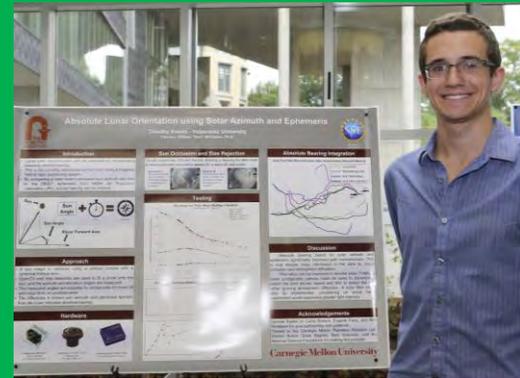
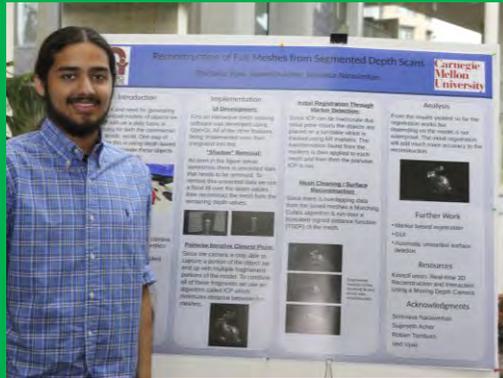
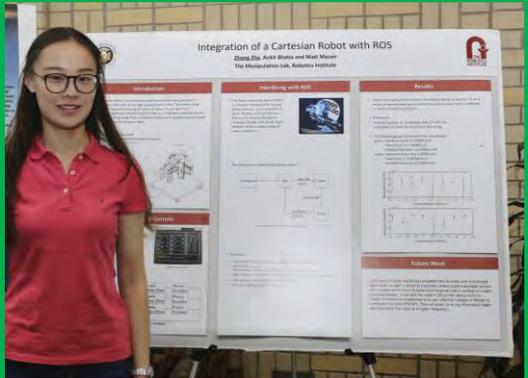
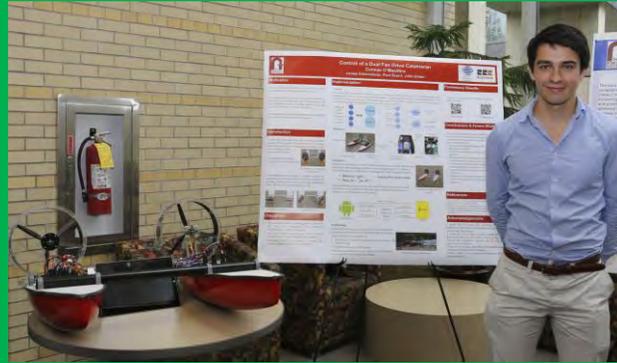
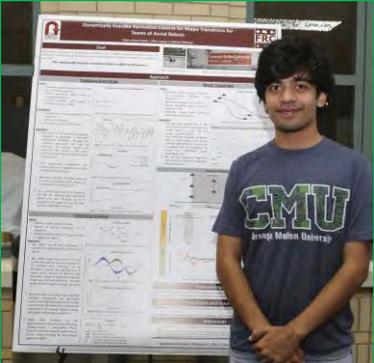
intelligent systems, and data science companies extend the scholars exposure from the robotics lab to product development, commercialization, and technology transfer.

Scholars benefit from RI's institutional experience from hosting undergraduate research programs for over nine years and an alumni network of over 200 previous undergraduate researchers now in graduate programs, academia, and industry both domestically and globally. In the fall of 2015, 33 RISS Program alumni will be attending CMU graduate programs in both Masters and PhD programs (29 within the School of Computer Science) or will be working as technical staff at the Robotics Institute. Also in the fall of 2015, a core group of RISS alumni are launching Pittsburgh-based alumni programming. This extensive community helps scholars successfully prepare for graduate school and maximize the summer experience. One of the program's strategic priorities is to extend access to robotics research opportunities to students from underrepresented groups and those from higher education institutions with fewer research opportunities.

Human capital is one of the most important resources driving today's knowledge economy. RISS connects a diverse group of talented undergraduate students from around the world to the Carnegie Mellon University community. The valuable contributions and connections that summer scholars make continue beyond this eleven week summer program.

PHOTO GALLERY

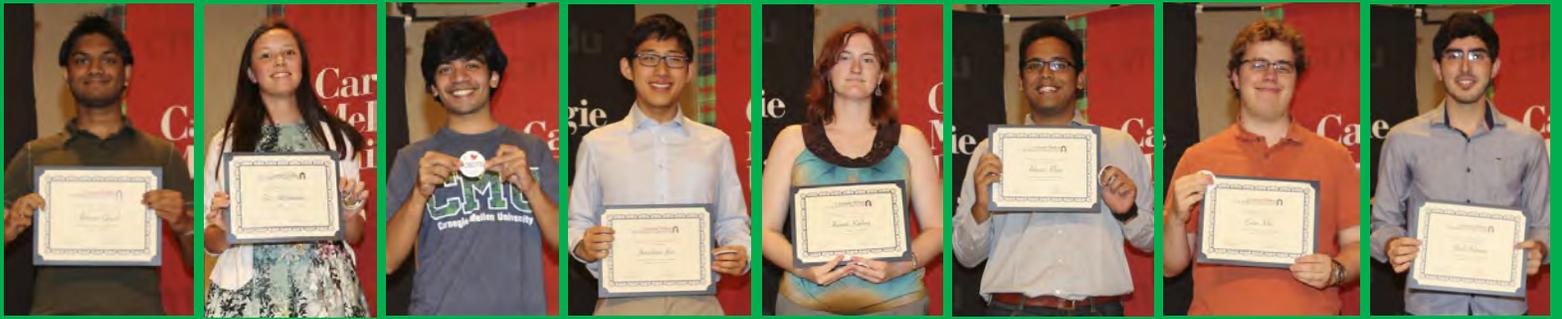
RISS 2015



RISS 2015 Presentations



RISS 2015 CEREMONY



Cohort RISS 2015



Working Papers

Authors



1. Sufyan Abbasi and Randy Sargent

Visualizing Air Quality: The Environmental Sensor Data Repository Explorer



2. Gabriela Amaral Araujo de Oliveira, Tony Dear, and Howie Choset

Development of an Online Learning Platform for an Undergraduate Robotics Course



3. Elias Bitencourt, Robert Paolini, and Matthew T. Mason

3D Vision with Kinect and Similar Sensors



4. Laura Brooks, Heather Knight, and Reid Simmons

Temporal Gestures for Expressive Motion



5. Nai Chen Chang, Xinlei Chen, and Abhinav Gupta

Differentiating Singularity and Multiplicity in Web Images



6. Felipe G. Galiza and David S. Touretzky

Analytical Inverse Kinematics of a 6-DOF Mantis Robot Arm and Modeling for Minimum Joints Load



7. Jonathan Y. Joo, Christopher Cunningham, Heather L. Jones, and William L. Whittaker

Waypoint Sequencing for Planetary Rovers with Significant Time-Dependent Energy Constraints



8. Ariana Keeling, Robert Paolini, and Matthew T. Mason

Autonomous Object Recovery in Manipulation Experiments



9. Arbaaz Khan, Shreyansh Daftry, and Martial Hebert

Multi Modal Pose Fusion for Monocular Flight



10. Ratnesh Madaan, Robert Paolini, Erol Sahin, and Matthew T. Mason

Predicting Orientations Under Manipulation Actions



11. Chirag Nagpal, Kyle Miller, Benedikt Boecking, and Artur Dubrawski

An Entity Resolution Approach to Isolate Instances of Human Trafficking Online



12. Ravenna Thielstrom, Heather Knight, and Reid Simmons

Generating Spatial Paths to Express Attentional Attitudes



13. Hadi Salman, Tony Dear, and Howie Choset

A Novel Skidding Model for a Snakeboard



14. Hadi Salman, Chaohui Gong, and Howie Choset

Shape Based Geometric Motion Planning for an Underactuated Highly-Articulated System



15. Minghan Wei

Detecting Vehicles Based On Smartphone-collected Images

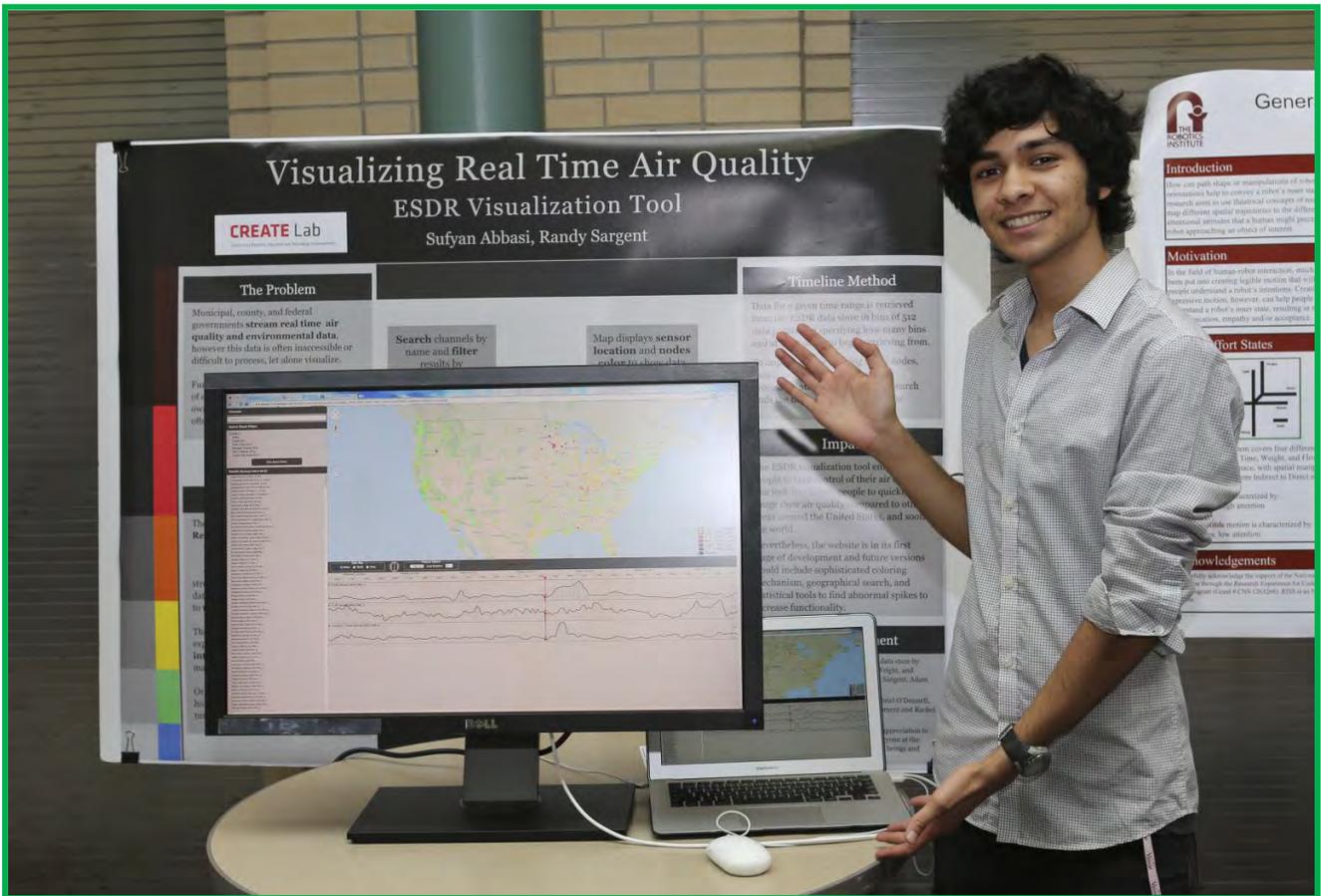


16. Xingchen Yu

Methods and Experiments of Pedestrian Detection

Sufyan Abbasi

RISS 2015



Visualizing Air Quality: The Environmental Sensor Data Repository Explorer

Sufyan Abbasi

Mentor: Randy Sargent

CREATE Lab, Carnegie Mellon Robotics Institute
Carnegie Mellon University

Abstract—Clean air is a basic human right and governments, from county to federal, as well as citizens possess public sensors that stream real time air quality and environmental data. However, this data is often inaccessible and lost over time. The Environmental Sensor Data Repository, developed by the CREATE Lab, is a server that collects, archives, and presents environmental sensor data. We developed a visualization tool in which users can explore ESDR temporally and spatially, specifically real-time, dynamic air quality data.

The ESDR tool empowers people to take their air quality back into their hands by presenting a story about air quality. This paper outlines the features of the ESDR Explorer website, how each feature is implemented, and future applications of the tool.

Index Terms—data visualization, air quality, ESDR

I. INTRODUCTION

PEOPLE are becoming more concerned about their air. Communities based around particularly noxious pollutants collect environmental data such as air or water quality that community members utilize as evidence of violations of environmental guidelines. Although this data may be archived locally, it is susceptible to being lost over time or not diffuse to larger audiences who might be able to mobilize for the community.

Established by the CREATE Lab at the Carnegie Mellon Robotics Institute, the Environmental Sensor Data Repository seeks to collect, archive, and present this data such that it is made available to the public around the world, as well as effectively gauge trends in environmental data on scales of weeks, months, and years. However, the original ESDR explorer tool was a simple website that only contained the list of all the sensor channels and when turned on, displayed graphs of the sensor data.

It was necessary to include a channel search and filter, a map of the sensor locations, and a method to visualize overall air quality on the map so that users of the website could easily access the trove of data relevant to them. The ESDR Explorer allows users to gauge their air quality now and over time while comparing it to other places around the nation. A map allows users to quickly find the nearest air quality sensors in their locale. In order to gauge air quality in context to the rest of the United States, nodes on the map color based on the EPA Air Quality Index and users may view trends in air quality between the many sensors.

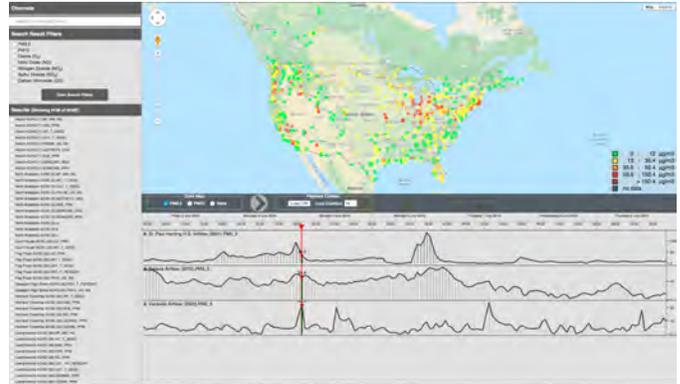


Fig. 1. ESDR Explorer website

This paper provides an overview of the website, outlines the implementation of each of the components, and provides insight into how such tools can be created for other such websites.

II. METHODS

A. Website Overview

The ESDR Explorer tool may be visited at esdr.cmucreatelab.org/browse, shown in Figure 1, which is a public link that is hosted by the CREATE Lab servers. The website is laid out into three section. A search and filter tool bar on the left side of the screen allows users to search for a particular channel based on keywords or filter for certain channels based on what the sensor data is reporting. A map of the sensor feeds (a collection of channels at the same location) displays nodes that can access the channels of a particular feed. Below that, a timeline and graphs open up when a channel is turned on to display the sensor data. Superimposed on the map is a tool bar which can be used to color the nodes based on values and toggle playback options.

During the page *onload*, the website requests the server for all current feeds in ESDR to load them into the “Results” section. Once all the feeds have been loaded, the browser initializes WebGL and the Google Maps API to render the nodes and the map onscreen respectively. At 30 MB/s, the website takes roughly five seconds to initialize a little under 9,500 channels (and counting). When idle, the website utilizes little CPU time despite running in WebGL.

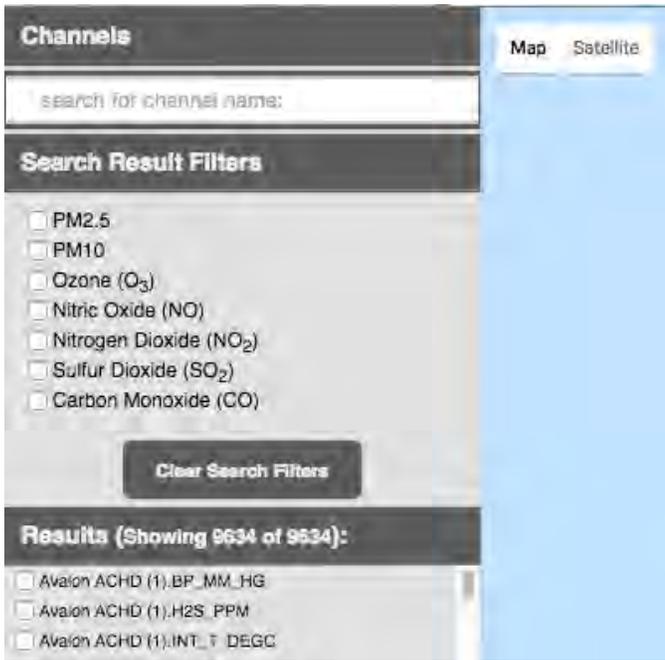


Fig. 2. Search and filter bar on ESDR Explorer website

In order to have a sensor display a user's environmental data, a user can create an account at esdr.cmucreatelab.org and follow the ESDR protocol from the terminal to stream the data in real time. Because the website pulls data from the server at every page reload, a node will appear on the map and their data accessible to the website. By either toggling the check box next to a channel name or clicking on the node on the map and selecting the check box from there, a graph is loaded underneath the map to display the data graphically that can be manipulated to show different time intervals.

B. Search and Filter Tool Bar

The first feature implemented was a search bar that allows users to find specific names of sensor channels. The search bar, shown in Figure 2, is case insensitive and can find partial matches of key words. Furthermore, spaces in the query act as key word separators with a conditional "and." The search bar shows the results of the query in the list of channel names as well as on the nodes of the map dynamically as the user types, foregoing the use of a search button. For example, the search "PM25 Avalon" would result in channel names that contain "PM25" and "Avalon" and show only the nodes on the map that satisfy those results.

In order to implement the search feature, jQuery selectors were appended in series with each key word between spaces and targets the input divs (check boxes) containing those key words. After selecting those input divs, all rest were hidden and only those shown. To hide the nodes on the map, a function was written in order to change the WebGL buffer for those resultant feed numbers to enabled, which will be discussed further in the map implementation.

Next, filters allow for shortening the large list of channel names to only display feeds and channels that contain a

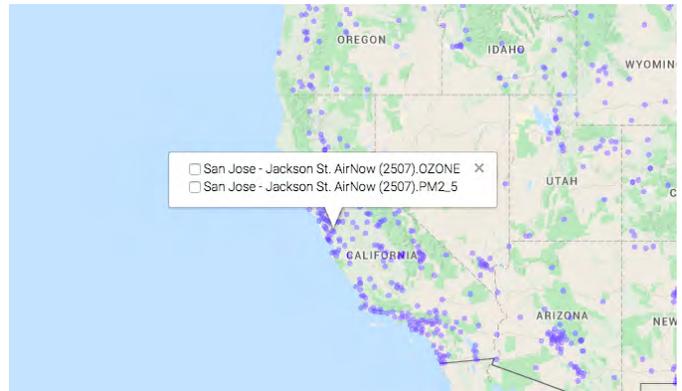


Fig. 3. Google Maps with WebGL implementation of nodes on ESDR Explorer

certain environmental data type, for example carbon monoxide or sulfur dioxide sensors. Although the ESDR server does not provide metadata for what kind of information is being displayed in each sensor; each feed is given a feed ID number which can be referenced and a channel name which contains keywords like "PM2.5" for particulate matter less than 2.5 microns or "SO2" for sulfur dioxide. At the initial page load, the channel name is categorized based on the naming conventions of the source of the data. For example, the EPA's AirNow categorizes all PM2.5 data as *PM2_5*, so their feed IDs are appended to an array for "PM2.5." When a filter check box is engaged, all feed IDs and channels within that category are displayed in the channel names and all other nodes are removed from the map. When subsequent filters are checked, the resultant channels include the other types checked.

Implementation of this works the same way as the search, where a jQuery selector for divs containing the *data* attribute for that channel type is concatenated together and the results of the search are shown and rest hidden.

C. Map

In order to render the map on screen, shown in Figure 3, the Google Maps API was employed and to draw the nodes on the map, a modified Canvas Layer was used in conjunction with WebGL. A WebGL buffer is generated on the initial page load that contains: the *xy* coordinates of the nodes to draw, a binary flag to indicate whether the node should be enabled or not after a search, the node's color in *rgba*, and the node's opacity when hovered, among other things.

When the map is zoomed or dragged, Google Maps returns an edited transformation matrix that the Canvas Layer uses to redraw the points at the new pixel locations. The node sizes are recalculated at each map zoom based on what level the map is zoomed at.

To implement the hover behavior of the nodes, the Google Maps API allows for a method to return the *xy* pixel coordinate of the cursor. The distances between this coordinate and every *xy* coordinate of the nodes are computed and the nodes with the smallest distances are recorded. If the node distance to the cursor is below a certain threshold (the radius of the node circle), that node's opacity value is increased in the

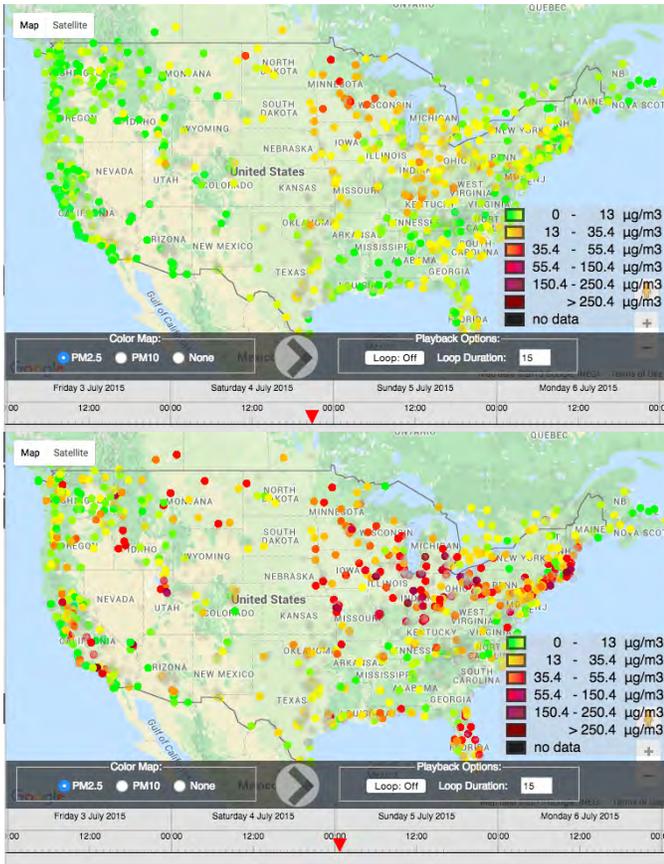


Fig. 4. Node coloration for July 4th, 2015 at 9:00pm EDT and July 5th, 2015 12:00am

WebGL buffer. Due to the fragment shader in WebGL, if two or more nodes are overlapping at this point, the opacity of each node is increased. For clicking on the node, the same distance calculations are made, but instead, a Google Maps information window is opened at the closest xy coordinate containing the input divs of the found feeds. If two or more nodes are overlapping at this point, the input divs of each feed is separated by a line.

D. Timeline

The most graphically and CPU intensive feature of the website is the timeline that recolors the nodes on the map based on EPA Air Quality Index colors as a user scrubs on the timeline, demonstrated in Figure 4. When a user clicks on, for example, “PM2.5” on the map tool bar, the timeline loads and a red cursor on the timeline appears which may be dragged. The nodes color for that specific time range.

The ESDR server can aggregate data together for many channels called *multifeeds* which returns a large JSON containing a list of timestamps and a comma separated list of values at that time. The website requests multiple *multifeed* data sets and processes them into an array of function calls may be iterated and executed to change the color of each node at a particular timestamp. The color is interpolated so that a gradient change in values can be shown based on the EPA Air Quality Index values for PM2.5 or PM10. Next, the timeline, implemented by

the CREATE Lab, contains a method that returns the position of the red cursor on the timeline. By determining the time position of the cursor as it is being dragged or played, a binary search is utilized to find the closest timestamp in the array and call the color changing functions at that time.

In order to make the colors more stand alone, all other nodes are reduced in opacity and turned gray.

E. Graphs

When a user checks a channel name, a plot opens up under the timeline displaying the data graphically. The graph, known as *Grapher*, was produced by the CREATE Lab to display real time data and has APIs that allow it to be easily customizable and accessible. Because this was originally implemented at the start of the project and not of my own, the implementation of this will not be elaborated upon in this paper.

III. FUTURE WORK

Although the website is in acceptable, working condition, there are still many features which could be implemented in order to increase the efficacy of the website. We are interested in adding an export button which allows users to export .csv files for opened channels to allow for statistical analysis of the data. To see trends in air quality, visualizing the wind speed and direction as vectors would give new meaning to the data. Finally, to be able to select any channel and define color parameters to interpolate would be an effective feature for users customize what kinds of data they want to visualize with color.

IV. CONCLUSION

In its current state, the ESDR Explorer tool visualizes trends in air quality that may not have been seen before. When, for example, a spike in air quality occurs, from fires to the Fourth of July, the map showcases how the pollutants in the air shift over time. By archiving the data and making it available to the public, people from community members, to scientists, to politicians have access to data that can be used as evidence to make lasting impacts.

We created this tool in order to tell stories with data, specifically environmental sensor data which directly correlates to people’s healths and wellbeing. By empowering people’s stories with data, or vice versa, we seek to change the world one sensor at a time.

ACKNOWLEDGMENT

The ESDR server was created by Chris Bartley, ESDR data store by Randy Sargent, Chris Bartley, and Anne Wright, and Grapher interface by Anne Wright, Randy Sargent, Adam Mihalcin, and Chris Bartley.

I extend my gratitude to Chris Bartley, Gabriel ODonnell, and Paul Dille for guidance in web development and Rachel Burcin for leading RISS.

Finally, I would like to extend my deepest appreciation to Randy Sargent, Illah Nourbakhsh, and everyone at the CREATE lab for being inspirational human beings and creating for humanity.

Gabriela Amaral Araujo de Oliveira

RISS 2015



Developing an Online Learning Platform for an Undergraduate Robotics Course



THE ROBOTICS INSTITUTE

Gabriela Amaral¹, Tony Dier², Howie Choset^{1,2}
¹Federal University of Lavras - Lavras MG, BR
²Carnegie Mellon University - Pittsburgh PA
 goliveira@compusernet.ufes.br | tonydier@cmu.edu | choset@cs.cmu.edu



SCIENCE
CENTERS FOR EXCELLENCE

Introduction

- Teaching robotics using web-based technologies supplements traditional classroom training.
- The use of a robot simulator in a robotics course enables the students to better understand the fundamental concepts by practical application.

Proposal: Development of an online learning platform for an introductory course, Robot Kinematics and Dynamics, based on Robotics Toolbox.

Robotics Toolbox Problems

The programming problems based on Robotics Toolbox explore classical topics in robotics, such as homogeneous transformations, forward and inverse kinematics, differential kinematics, and dynamics.

Problem 1: Finding the visualization of a robot using a program rather than inspection.



Figure 2. 1. Work Jacobian of a 2-link Planar. In the figure, the second link is bigger than the first. In the figure 3, the second link has the same size.

Problem 2: Finding multiple solutions for the inverse kinematics problem through visualization.



Figure 4, 5. Inverse Kinematics Solutions. In the figure 4, there is a right-handed solution. In the figure 5, there is a left-handed solution.

Problem 3: Finding a new way of understanding the columns of the Jacobian through small changes in one joint coordinates that affect the pose of the end-effector.



Figure 6, 7. Jacobian solution before and after small changes in one joint coordinates. One of the important contributions is being able to visualize these problems, as opposed to before which was just solving them on paper.

Video-Based Lectures

- The lectures are taught in class using PowerPoint slides.
- The course also offers video-based lectures recorded outside the classroom, which are available via Panopto video platform software.
- Some of our pedagogical goals are: avoid making class sessions too redundant with video recordings and reduce the amount of time students spend watching videos.
- The new videos are 43% shorter, in terms of both average and total length, than the old videos recorded inside the classroom.



Figure 8. Comparison between the total length of new and old videos per week.



Figure 9. Comparison between the average length of new and old videos per week.

Methods

Comparison of open-source robot simulators for teaching robotics.

Simulator	Description
Robotics Toolbox	Robot Simulator with integrated development environment. It offers an educational license.
Arboreo	3-D simulator for populations of robots in complex indoor and outdoor environments.
Robotics Toolbox	Software package that allows a Matlab user to develop datatypes fundamental to robotics.

Table 1. Comparison among robot simulators.

Implement the current homework and laboratory assignments using Robotics Toolbox for Matlab.

Generate new examples for the course based on the Robotics Toolbox.

Improve the video-based lectures in terms of quality, educational load, and pedagogical benefits.

Use surveys conducted with students of the course and develop assessment methods in form of review and quiz questions.

Conclusion

Our Robotics Toolbox problem development and video-based lectures will be important aspects in building the online learning platform for introductory robot kinematics.

Future work

- Implement the Robotics Toolbox problems in the next Fall semester.
- Connect Robotics Toolbox to a 3D simulator such as V-Rep and Gazebo.
- Develop a public online course in a platform known worldwide.

References

Carke, Peter. "A robotics toolbox for MATLAB." *Robotics & Automation Magazine*, 1998, 24-32.

Robot Kinematics and Dynamics. Retrieved from <http://robotkinematics.org>. Accessed in May, 2015.

24

Development of an Online Learning Platform for an Undergraduate Robotics Course

Gabriela A. A. de Oliveira, Tony Dear, Howie Choset

Abstract—The teaching and use of robotics have come a long way in recent years, starting from university engineering departments developing their own robots to the introduction of simple consumer-based robots to universities down to the primary school level. With these developments, an important issue came into focus: what is the most effective way to approach teaching robotics to students? To address this issue, we propose the development of an online learning platform for an undergraduate course, Robot Kinematics and Dynamics. This platform will be based on the use of Robotics Toolbox for MATLAB, a tool that has been gaining popularity in education. This has allowed us to supplement traditional paper-and-pencil problems with educational programming problems on Toolbox, covering topics such as homogeneous transformations, forward and inverse kinematics, differential kinematics, and dynamics. In addition to programming assignments, our platform includes the development of various form of media such as video-based lectures, which enable the students to better understand the theoretical concepts behind introductory robotics through simulation and visualization.

Index Terms— Introductory robotics, online learning platform, robotics course, robot simulation, robot kinematics, robot dynamics, Robotics Toolbox for MATLAB, video-based lecture, undergraduate course.

I. INTRODUCTION

ROBOTICS is an essential topic in the current technological world, and consequently the teaching of robotics has seen growth in recent years. This ranges from university engineering departments developing their own robots, such as LEGO Mindstorms, to universities down to the primary school level. From a broad perspective, robotics is a multidisciplinary subject that requires knowledge of a numbers of disciplines such as kinematics, dynamics, control, mechanics, and electronics engineering. Traditional classroom methodologies are often inadequate in helping students understand the complex theoretical concepts underlying

robotics. The main challenge in teaching robotics to students has been in effectively coming up with the appropriate content and complexity to improve learning.

In the current engineering education paradigm, the use of simulation tools has being fundamental for students' motivation and understanding of the course content through programming and simulation. The work realized by the Department of Computer Science and Systems Engineering at University of Zaragoza, Spain, presents a remarkable success of courses based on simulation tools with Control and Programming of Robots, as well as Industrial Robotics. The results of this research show that the final grades covering from year 2003 to 2009 were improved and the percentage of students who fail to pass was reduced [1]. Moreover, research conducted by the Electrical Engineering and Computer Science Department at the United States Military Academy about teaching problem solving with robots found that the learning with simulation is much easier and faster for an introductory course, while maintaining cost and convenience advantages [2].

Robotics Toolbox is a software package for MATLAB extensively used in universities for teaching and research in the robotics field through interactive simulation [3]. The Mechanical and Aerospace Engineering Department at Princeton University has used Robotics Toolbox in their undergraduate course, Robotics and Intelligent Systems [4]. Another university that has used Robotics Toolbox for teaching Introductory Robotics is the Engineering Department at the University of FEI in Brazil.

With the development of web-based technologies, universities have been exploring Internet resources to communicate and collaborate in an educational context. These technological resources represent a new method of transmitting university education, and complement traditional teaching approaches for undergraduate students [5]. MOOCs, or massive open online courses, are part of this group of resources because they allow anyone with an Internet connection to enroll in a course, are credit-less, and are usually free [6]. Supplementing a classroom course with online tools provides increased opportunities for students to view and review course materials and therefore offers them better retention of the learned content. Online learning platforms make a significant impact in higher education today and, as technology evolves, promise to deliver even greater benefits in the future.

In this context, we propose the development of an online

“This work was supported in part by the Science Without Borders United States and CAPES (Coordenação de Aperfeiçoamento de Pessoal de O. A. A. Gabriela is with Federal University of Lavras, MG 357000 Brazil, undergraduate student, in an exchange program at Temple University, PA 19121 USA. She is a 2015 Summer Scholar Alumni with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15289 USA (e-mail: goliveira@computacao.ufla.br).

D. Tony is with the Robotics Institute, Carnegie Mellon University, PhD student, PA 15289 USA (e-mail: tonydear@cmu.edu).

C. Howie is professor with the Robotics Institute, Carnegie Mellon University, PA 15289 USA (e-mail: choset@cs.cmu.edu).

learning platform for introductory robotics, which includes the development of multiple forms of media, including video-based lectures, programming and simulation assignments, and assessment methods in the form of review and quiz questions. In this paper, we will focus mainly on presenting a new method of design and implementation of educational assignments in a junior- and senior-level undergraduate course, 16-384: Robot Kinematics and Dynamics, at the Robotics Institute at Carnegie Mellon University (CMU). We propose the implementation of programming problems based on Robotics Toolbox for MATLAB that explore classical topics in robotics, such as homogeneous transformations, forward and inverse kinematics, differential kinematics, and dynamics. Some simulation problems using the robotics toolbox are presented here. The simulation tool, used in this course, is essential for the students to better understand the theoretical concepts behind introductory robotics.

The paper is organized as follows. The second section, named “Methodology”, describes the processes used during the development of the research. The third section, named “Results”, is the main section of this paper and presents some examples of Robotics Toolbox assignments proposed for our kinematics course. The video-based lectures are also presented in the third section, which describes the primary pedagogical benefits of the videos for our introductory robotics course. Conclusions and future work are discussed in the fourth section.

II. METHODOLOGY

The current robot kinematics and dynamics course consists of lectures, which are taught using PowerPoint slides and videos, paper-and-pencil homework problems, and two short labs with actual robots. We propose the development of an online learning platform with video-based lectures and programming assignments based on simulation tools.

A. Studying the course content

The robot kinematics and dynamics course focuses on kinematics, dynamics, and programming of robotic mechanisms. Arms are the primary application and classical robots such as the Denso and Puma 560 are studied during the classes. This course assumes that the students have a prior knowledge, acquired in prerequisite courses, about calculus, basic programming, and algebra, like linear matrix.

The course has a weekly schedule in order to cover a particular set of topics along with learning goals. The main effort prior to starting the development of programming assignments was to study the video and slide content week by week as well as to work through some of the current paper-and-pencil homework problems.

The contents studied per week are presented in Table I.

Week	Content
Week 1	Degrees of freedom; concepts of joints and linkages; rotation matrices for 2D and 3D rotations.
Week 2	Matrix composition; parameterization of rotation matrices, covering the Euler angle, yaw-pitch-roll, and the angle-axis formulations.

Week 3	Homogeneous transformation.
Week 4	Denavit-Hartenberg (DH) convention; examples of DH for different arm robots; forward kinematics.
Week 5	Inverse kinematics; numerical inverse kinematics.
Week 6	Derived and examples of the Jacobian; differential kinematics.
Week 7	Inverse differential kinematics; singularities.
Week 8	Skew-symmetric matrices; angular and linear velocities.
Week 9	Manipulability for mechanical systems; dynamics of rigid bodies; standard form of the equations of motion.
Week 10	Coriolis terms of the equations of motion; state space form; and rigid bodies dynamics with friction.

B. Comparing Robot Simulators

Simulators are an important tool in robotics research and teaching. Considering the increasing number of open-source robotic simulation tools, we compared the most efficient robotic software for teaching based on the research developed by the ASTRA Robotics Lab at the University of Texas, Arlington [7]. Their survey paper presents a detailed overview among the most recent and popular open-source robotic software for simulation.

After analyzing the features offered by all software studied by the above on the paper, the robotics simulator software presented in Table II were further investigated. The main criteria used to select these four robot simulators were to be 1) open-source; 2) available for Mac, Linux, and Windows; 3) easy to install; and 4) appropriate for robotics related education.

Simulator	Description
V-Rep	Robot simulator with integrated development environment. It offers an educational license, V-Rep pro edu.
Gazebo	3D simulator for populations of robots in complex indoor and outdoor environments.
Robotics Toolbox for MATLAB	Software package that allows a MATLAB user to develop datatypes fundamental to robotics.
Player/Stage	Player is a robot device interface to a variety of robot and sensor hardware. Stage is a multiple robot simulator for robots moving in and sensing a two-dimensional bitmapped environment.

Robotics Toolbox for MATLAB was developed to enable teachers and students to better understand the theoretical concepts behind introductory robotics through easy simulation. This software package has been extensively used in universities for teaching and research in the robotics field [8]. Based on that, we chose Robotics Toolbox as the tool to develop programming assignments in the robot kinematics and dynamics course.

C. Robotics Toolbox homework problems

The programming problems based on Toolbox explore classical topics in robotics, such as homogeneous transformation, inverse and forward kinematics, differential kinematics, and dynamics. We did some of the current homework problems and labs, but instead of using paper and pencil, the solutions were found using robotics toolbox

through programming and simulation. In addition to that, new assignment problems based on robotics toolbox were developed. The Robotics Toolbox examples enable the students to visualize the problems, as opposed to the previous method of solving on paper.

D. Surveys with students

The robot kinematics and dynamics course is offered to CMU students every fall semester. Surveys conducted with these students during the semester 2014 were analyzed in order to learn some of the issues that students had about the course and improve the course material and organization based on that. In that time, the surveys were carried out weekly, which enabled us to see the common questions among the students for particular topics. We plan to create new assessment method in form of review material or quiz questions, and redo the PowerPoint slides as well, to cover the topics that cause most questions in the current course format.

The students provided some suggestions for the improvement of the video-based lectures such as dividing a long video into short video segments, while still upload the slides as a PDF or PowerPoint presentation on the online platform. They also opined for the possibility of covering examples, derivations and proofs, and prerequisite material using videos. Because some weekly materials are harder than others, an additional suggestion was made to reorganize them by level of difficulty. In addition to that, the students used the video-based lectures as review for working on the homework and studying for the exams. In order to solve the students' issues, their suggestions will be considered for the new video recording session.

E. Video-based Lectures

The current videos are available via Panopto Video Platform Software, which enables simple video editing and, more essentially, allows for PowerPoint slides to be shown adjacent to the video. It enables the students to have a clear outline of the lectures, take notes on individual slides, and search keywords.

While studying the video and slides, we noted the quality, length, and method of videos in order to improve the video-based lectures and improve their pedagogical effectiveness. Some videos had bad lighting, which can be a problem because the viewer may have difficulty reading the slides presented in the video and loses interactivity with the professor who is talking in a dark place. Another issue was the timing of the videos. Some individual videos of each week are longer than others. For example, the smallest video is about two minutes, whereas the longest one is about thirty minutes. For the subsequent versions of these videos, the timing will be pared down in a maximum of fifteen minutes each. Also, the quality of the video will be a major consideration.

III. RESULTS

The online learning platform, currently in progress, will offer the course's syllabus and schedule, Robotics Toolbox homework problems, weekly video-based lectures, and a form for video feedback.

A. Simulation tool and Robotics Toolbox homework problems

Robotics Toolbox for MATLAB provides many functions that are required in robotics and address the kinematics and dynamics areas. The Toolbox is useful for simulation and analysis of robotics problems. Two main Robotics Toolbox classes were explored to develop the assignments problems, Link and SerialLink. The class SerialLink represents the kinematics and dynamics of serial-link manipulators by description matrices. The DH parameters of the robot can be created for any serial-link manipulator [9].

In the online learning platform, two main learning aspects can be benefited from the use of simulation tools in the context of robot kinematics related subject: visualizations of robot configurations and the acquisition of robot programming skills. The main learning improvements provided by the use of Robotics Toolbox homework problems are the in-depth understanding of the homogenous transformations and Denavit Hartenberg (DH) convention, and the better comprehension of the inverse and forward kinematics problems and the Jacobian problem.

The Robotics Toolbox homework problems are presented to the students as programming scripts. For example, given the arm link length and joint limits, find the workspace of the robot. To achieve this, students follow the DH notation, which is an input of the algorithm together with the joint limits. The DH convention [10] defines a coordinate system attached to each link of a chain, allowing determination of the forward kinematics of the complete chain in a systematic way. The solution for this problem is based on programming rather than inspection and has as output the plotting of the robot workspace, as shown in the Figure 1 and 2.

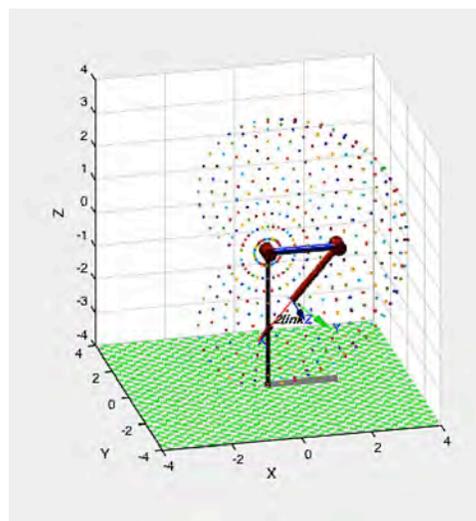


Fig. 1. Workspace of a two-link robot, where the length of the two link is the same.

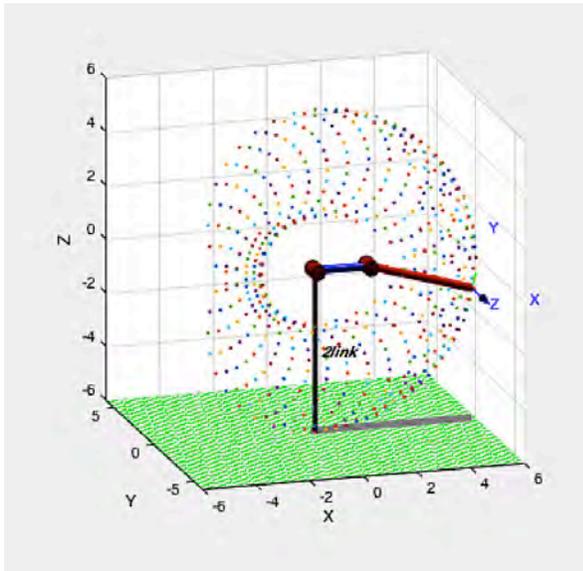


Fig. 2. Workspace of a two-link robot, where the length of the first link is smaller than the second link.

In the problem of finding the robot workspace we use the Robotics Toolbox classes `Link` and `SerialLink`, in addition to the method `plot3`, which was used to plot the robot arm.

Forward kinematics is a problem of determining the position and orientation of the end effector given the values for the joint variables of the robot. The inverse kinematics problem is to determine the values of the joint variables given the end effector's position and orientation. One example extension of this latter problem is to find multiple solutions for the inverse kinematics problem through visualization, where the input is the DH matrix and the joints angle configuration, and the output is the plotting of at least two different solutions for this problem, as shown in the Figures 3 and 4. For this problem, the Robotics Toolbox class `SerialLink` and the methods `fkine`, `ikine`, and `plot` were used. “`fkine`” is a forward kinematics method to determine the pose of the robot end-effector. “`ikine`” is a numerical inverse kinematics method that presents the joint coordinates corresponding to the robot end-effector pose. Finally, “`plot`” is a method used to plot the robot arm.

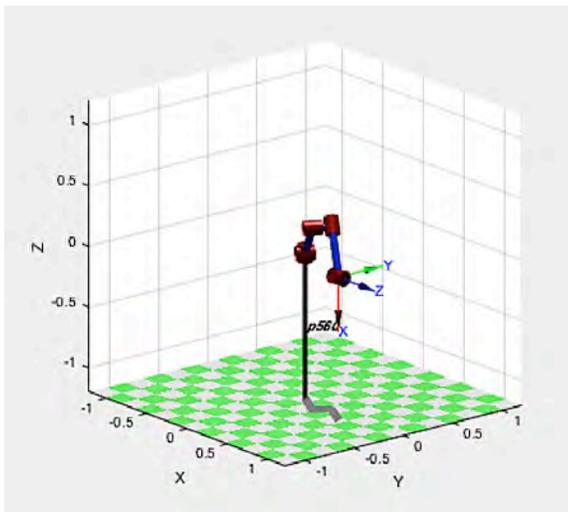


Fig. 3. Left-handed solution for the inverse kinematics problem.

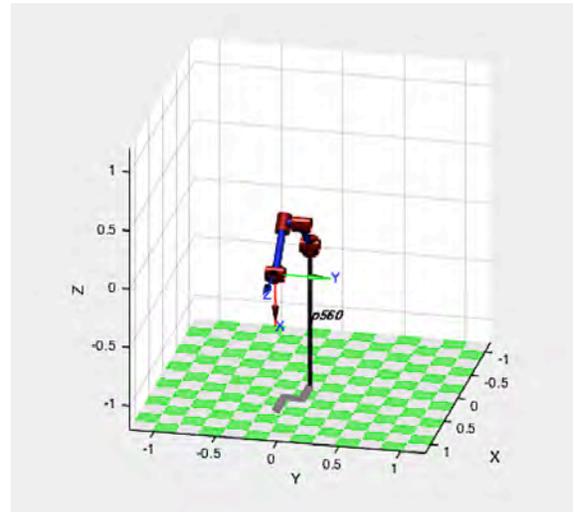


Fig. 4. Right-handed solution for the inverse kinematics problem.

A learning goal of the Jacobian concept is to understand the columns of the Jacobian through small changes, for example *magnitude* 10^{-2} , in particular joint coordinates, which then affect the pose of the end-effector. The Jacobians are vector versions of derivatives that are written as $\partial f/\partial x$ and $\partial f/\partial y$, where f is a function of a vector $f: \mathbb{R} \rightarrow \mathbb{R}$. In this problem, the input is the DH matrix and the joints angle configuration, and the output is the plotting of the robot arm in the initial joint configuration and after small changes in joint coordinates. In this problem, we use the Robotics Toolbox methods `J` (Jacobian), `jacob0`, and `plot`.

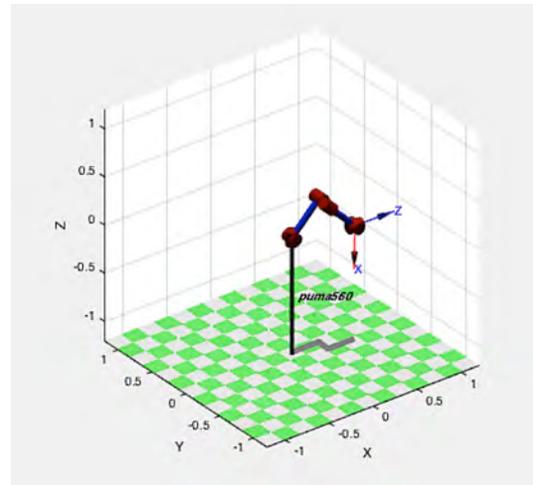


Fig. 5. Jacobian problem with the initial joint coordinates.

B. Video-based lectures

For our video-based lectures, some of our pedagogical goals are to reduce the amount of time students spend working through the content, because of the extra workload of watching videos, and avoid making videos redundant with the class sessions. In order to achieve these goals, the video-based lectures are recorded outside of the student-occupied classroom in a quiet setting, in which the professor explains the PowerPoint presentation. These videos are shorter than the past videos recorded inside the classroom because the focus is

the slide content, without student-initiated pauses to answer questions or give examples on the blackboard. In the past, the video recording of the blackboard presented many issues for the user such as illegible writing and mistakes that were difficult to fix in post-processing.

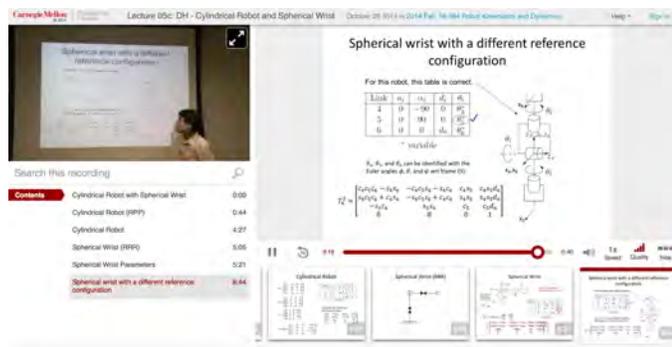


Fig. 6. A video-based lecture on the online learning platform.

The new videos are 43% shorter, in terms of average length, than the old videos of the blackboard inside the classroom with the students. The Figure 7 shows a comparison between the average length of the new and old videos per week. The course has a total of 13 weeks and each week has a different number of video-based lectures. For example, week 2 is about rotation matrices and has one video, whereas week 5 presents the topic Denavit Hartenberg convention and has five videos. Organizing the weekly videos in parts enables the students to take a short break before watching the subsequent video and also helps them to find specific class topics. If a student has questions about an example of a specific weekly content, he can go through the content of one small video only, instead of reviewing the whole weekly content recorded in a big video.

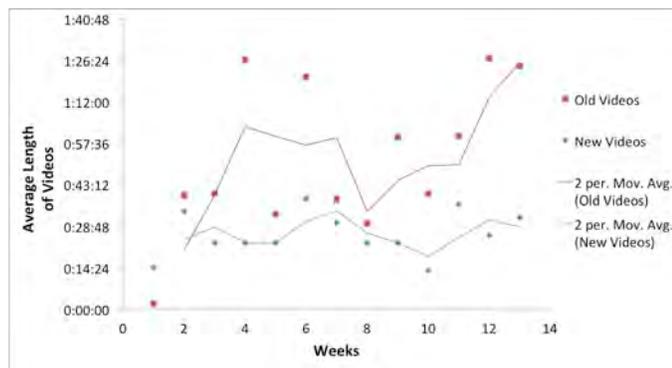


Fig. 7. Comparison between the average length of the new and old videos per week.

IV. CONCLUSION

In this work, we describe an online learning platform for an introductory robotics course that takes advantage of a simulation tool, Robotics Toolbox for MATLAB. The programming problems using Toolbox together with the video-based lectures will be important aspects in building the online learning platform for the robot kinematics course. The course with the Toolbox homework problems will be applied in the next fall semester, which will enable us to analyze the students' performance in learning based on simulation tools

and compare it with the traditional teaching method.

In order to perform the course and guarantee success, new Robotics Toolbox problems will be elaborated to better understand the theoretical concepts behind the introductory robotics. A future goal is to connect the Robotics Toolbox scripts with a 3D simulator such as V-Rep Pro Edu, which enables the visualization of the problems in a real robot and also allows the development of more advanced problems.

When the video-based lectures and Robotics Toolbox problems are ready, we plan to develop a public online course in a platform offered worldwide.

ACKNOWLEDGMENT

A. A. O. Gabriela thanks the Robotics Institute Summer Program for the opportunity of being a 2015 Summer Scholar. She also thanks the Robotics Institute's Biorobotics lab for all support through the summer research project, in particular, the PhD student Tony Dear and the professor Howie Choset.

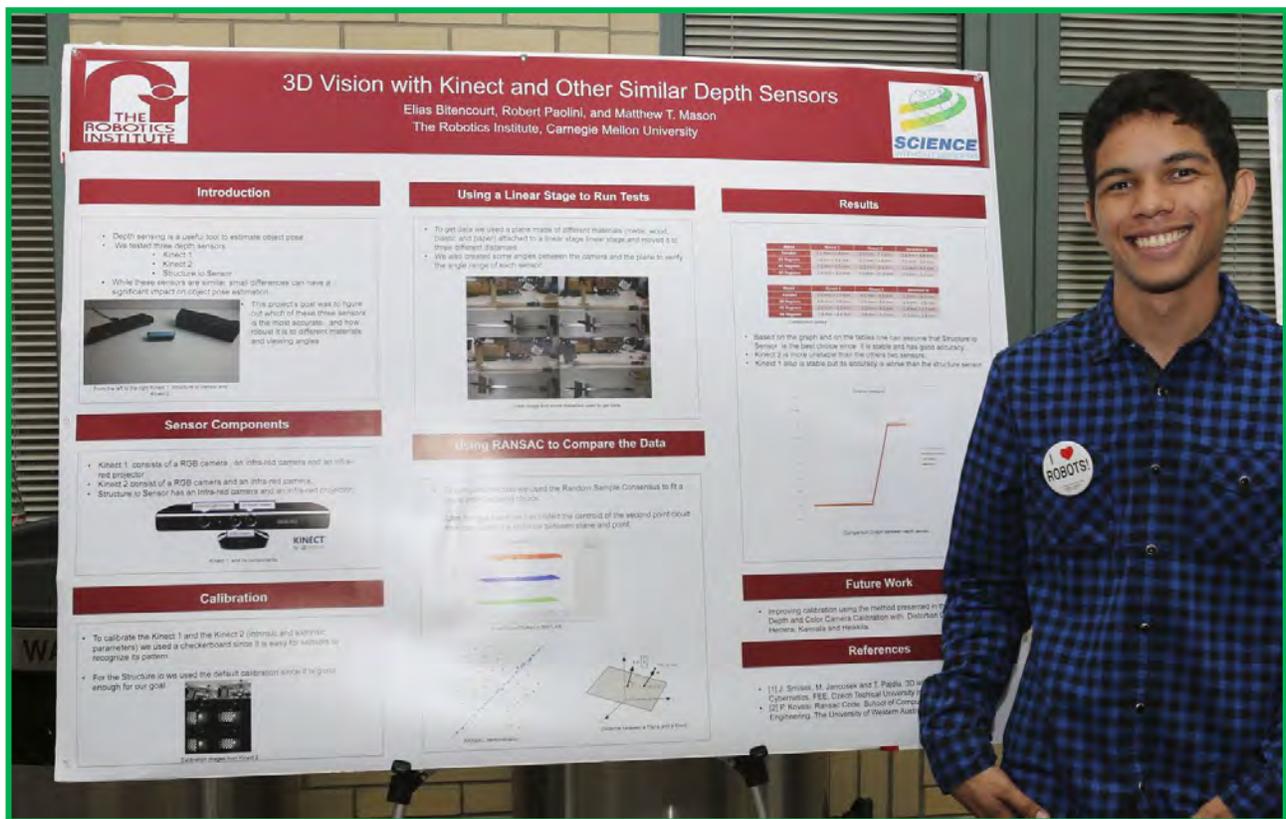
REFERENCES

- [1] López-Nicolás, G. A. Romeo, and J. J. Guerrero. "Active Learning in Robotics based on Simulation Tools." *Computer Applications in Engineering Education* 22.3 (2014): 509-515.
- [2] Flowers, Thomas R., and Karl A. Gossett. "Teaching problem solving, computing, and information technology with robots." *Journal of Computing Sciences in Colleges* 17.6 (2002): 45-55.
- [3] Corke, Peter. "A robotics toolbox for MATLAB." *Robotics & Automation Magazine, IEEE* 3.1 (1996): 24-32.
- [4] Robotics and Intelligent Systems. Retrieved from: Available: <<http://www.princeton.edu/~stengel/MAE345.html>>. Accessed in July, 2015.
- [5] Jara, Carlos A., Et Al. "Hands-on Experience of Undergraduate Students in Automatics and Robotics using a Virtual and Remote Laboratory." *Computers & Education* 57.4 (2011): 2451-2461.
- [6] L. Pappano. "The Year of the MOOC," *The New York Times*, vol. 2, no. 12, 2012.
- [7] Staranowicz, Aaron, and Gian Luca Mariottini. "A survey and comparison of commercial and open-source robotic simulator software." *Proceedings of the 4th International Conference on Pervasive Technologies Related to Assistive Environments. ACM*, 2011.
- [8] Corke, Peter. "MATLAB toolboxes: robotics and vision for students and teachers." *IEEE Robotics & automation magazine* 14.4 (2007): 16-17.
- [9] Corke, Peter I. "A computer tool for simulation and analysis: the Robotics Toolbox for MATLAB." *Proc. National Conf. Australian Robot Association*. 1995.
- [10] J. Denavit and R. S. Hartenberg, A kinematic notation for lower- pair mechanisms based on matrices, *Trans ASME J Appl Mech* 23 (1955), 215-221.
- [11] Robot Kinema-cs and Dynamics. Retrieved from <<http://robotkinema-cs.org>>. Accessed in May, 2015.
- [12] Serial Link Robot Class. Retrieved from <<http://www.petercorke.com/RTB/r9/html/SerialLink.html>>. Accessed in June 2015.

- [13] Robot Manipulator Link Class. Retrieved from <<http://www.petercorke.com/RTB/r9/html/Link.html>>. Accessed in June 2015.
- [14] Corke, Peter. Robotics, vision and control: fundamental algorithms in MATLAB. Vol. 73. Springer Science & Business Media, 2011.
- [15] Solved Problems in Robotics Kinematics using the Robotics Toolbox. Retrieved from <<http://www-iri.upc.es/people/thomas/papers/PROBLEMS.pdf>>. Accessed in July 2015.
- [16] V-REP, Virtual Robot Experimentation Platform. Retrieved from <<http://www.coppeliarobotics.com>>. Accessed in June 2015.
- [17] Gazebo, Robot Simulation Made Easy. Retrieved from <<http://gazebosim.org>>. Accessed in June 2015.
- [18] The Player Project. Retrieved from <<http://playerstage.sourceforge.net>>. Accessed in June 2016.

Elias Bitencourt

RISS 2015



3D Vision with Kinect and Similar Sensors

Elias Bitencourt, Robert Paolini, and Matthew T. Mason

The Robotics Institute, Carnegie Mellon University

ABSTRACT - We analyze three different depth sensors, Kinect 1, Kinect 2 and structure.io, to figure out which one is the best to estimate object pose. The method is design to have three main points: accuracy, types of material and the angles that the sensor can recognize. The test requires a linear stage with a planar surface moved in different distances and in different angles then take the point clouds. After recording point clouds, we use RANSAC to fit a plane. In our experiments, structure.io shows the best accuracy in different types of materials and angles.

Index terms - Kinect, 3D sensing, depth camera, RGB-D camera

data and do depth measurements. Kinect 360 (fig. 1) consist of a RGB camera, an infra-red sensor and an infra-red projector. The Kinect One has a HD RGB camera and an infra-red camera.



Figure 1: Kinect 360 and its components

1. INTRODUCTION

Since its invention, Microsoft Kinect has become popular in the scientific community, mainly in computer vision community. Besides Microsoft Kinect there are other sensors, such as Structure.io, that have been becoming popular in this field. The primarily reason to create the Microsoft Kinect and the others was to use it to electronics games and human interface, and 3D reconstruction. However because their low cost and satisfactory accuracy they have been used in other fields such as robot manipulation and navigation, and general 3D sensing.

In some fields of application the required accuracy has to be stronger, because even it being good sometime it isn't satisfactory. As examples of these fields that need a better accuracy one can cite robotics and similar areas which have been investigating how to improve the sensors accuracy and which one has the best accuracy.

In order to improve the calibration we tested three different sensors (Kinect 360, Kinect One and Structure.io) and figured out which is the best to estimate a pose object in a range 1.5 meters.

2. SENSOR COMPONENTS

The sensors have similar hardware. All of them have a depth cameras that allow them get depth

3. CALIBRATION METHOD

The first step before collect data is calibrate the sensors in order to improve the manufactory calibration. The used method to do so in some of our sensor was the Checkerboard Method since it has a pattern that the sensor can recognize easily. This method consist of positioning a checkerboard in front a sensor and move it around the frame corners as is showed in figure 2. However we didn't calibrate the Structure.io since its default calibration showed us results good enough. We used thereabout five frames for each corner to calibrate Kinect 360 [1] and 30 for Kinect one.

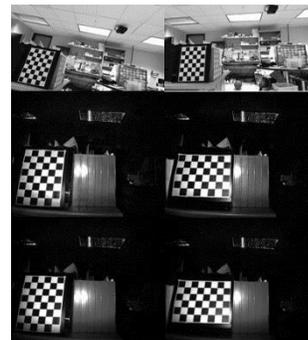


Figure 2: IR images from Kinect One

4. USING A LINEAR STAGE TO COLLECT DATA

In order to compare the sensors we used a linear stage to move a flat surface and get images in three different distances (30cm, 60cm and 90cm). To operate we used ROS (Robot Operating System). Besides changing the distances we also changed the type of material and the camera angle in order to make sure to know which sensor is the widest sensor, since for our goal not only the accuracy matters but also how it works in different situations. We got twelve point clouds for each type of material in each sensor, totalizing sixty points per sensor and one hundred eighty total.



Figure 3: Plane in different distances and angles

5. USING RANSAC TO COMPARE THE DATA

After collecting all our data, we had to find a way to compare the data obtained from each sensor. The method used to compare the data was RANSAC (Random Sample Consensus). This method takes two points then draw a line and see which points are inliers of the two first points as is demonstrated in figure 4.

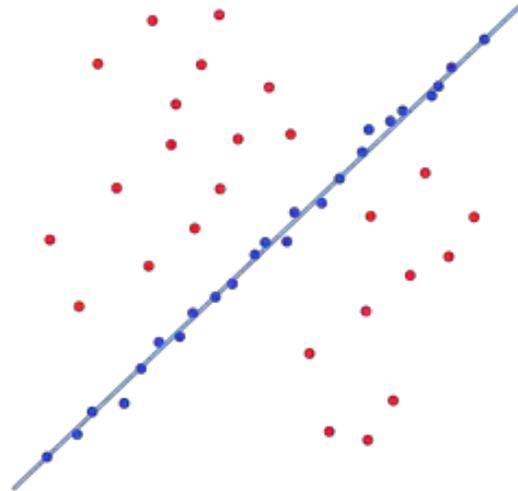


Figure 4: How RANSAC works

We put our data into a RANSAC code [3] and it return the distance between the planes. However we set up the distance that one random point should be from the line to be considered inliers. We set 0.007 meters when the plane was moved 30 cm, and 0.001 meters when the plane was moved 60 cm and 90cm. This difference between the first distance and the others is because these values returned more reliable data.

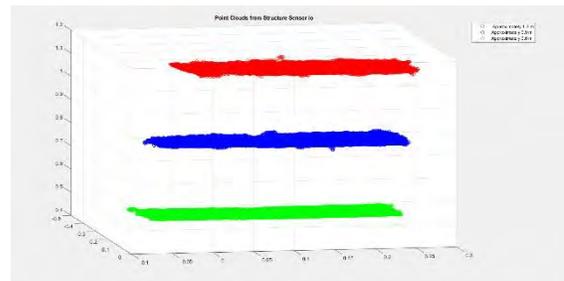


Figure 4: Planes plot

6. RESULTS

To demonstrate the results we decided use the data obtained from wood and metal, since the sensors recognize the wood easily and the metal is the hardest type of material recognized for our sensors. Based on the figure 5 one can assume that Structure.io is the best choice.

The results were surprising since the Microsoft Kinect One had the worst results. It was supposed to be the sensor with the best accuracy because its RGB camera is HD and it's a Kinect 360 upgrade. Its results are unstable and not reliable. As showed in table 1 the error variation is too large and one cannot trust it. The Kinect 360 had stable results but the error was bigger than the Structure.io. And Structure.io is the most promisor sensor for our goal, because it showed the smallest error margin. Moreover, its calibration can be improved even more since was used its default calibration.

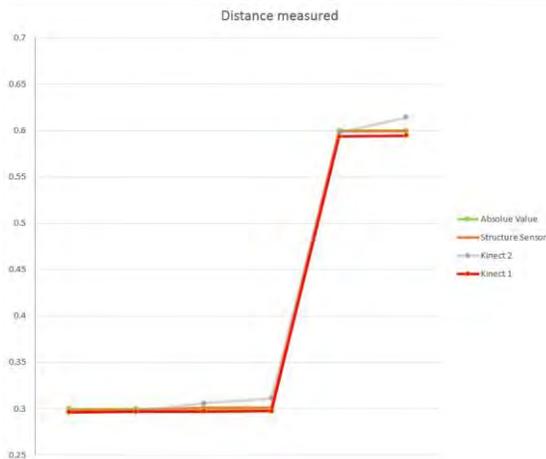


Figure 5: Trials

Metal	Kinect 1	Kinect 2	Structure io
Parallel	2.1 mm – 5.8 mm	3.6 mm – 7.3 mm	1.6 mm – 4.8 mm
30 Degrees	1.4 mm – 4.3 mm	2.7 mm – 3.8 mm	0.2 mm – 4.9 mm
45 Degrees	3.1 mm – 6.6 mm	1.3 mm – 6.4 mm	1.1 mm – 4.5 mm
60 Degrees	1.9 mm – 6.3 mm	2.9 mm – 2.19 cm	0.1 mm – 0.6 mm

Table 1: error range for each sensor

Wood	Kinect 1	Kinect 2	Structure io
Parallel	3.9 mm – 7.5 mm	4.1 mm – 9.9 mm	1.9 mm – 4.9 mm
30 Degrees	0.4 mm – 3.0 mm	1.3 mm – 3.3 mm	2.0 mm – 5.8 mm
45 Degrees	1.6 mm – 4.2 mm	1.1 mm – 3.3 mm	0.8 mm – 2.5 mm
60 Degrees	1.8 mm – 4.6 mm	0.8 mm – 4.0 mm	0.7 mm – 2.8 mm

Table 2: error range for each sensor

7. CONCLUSION

This paper has demonstrated that Structure.io is the best depth sensor for estimate the object pose within a range of 1.5 meters. In addition, the sensor recognized all the materials used in our tests. Then one can assume that it will work for our purpose.

8. ACKNOWLEDGEMENT

This work was supported in part by the CAPES, Higher Education Personnel Improvement Coordination.

All the analysis conducted in this work has been performed in Matlab environment, part using Ransac Code [3].

9. REFERENCES

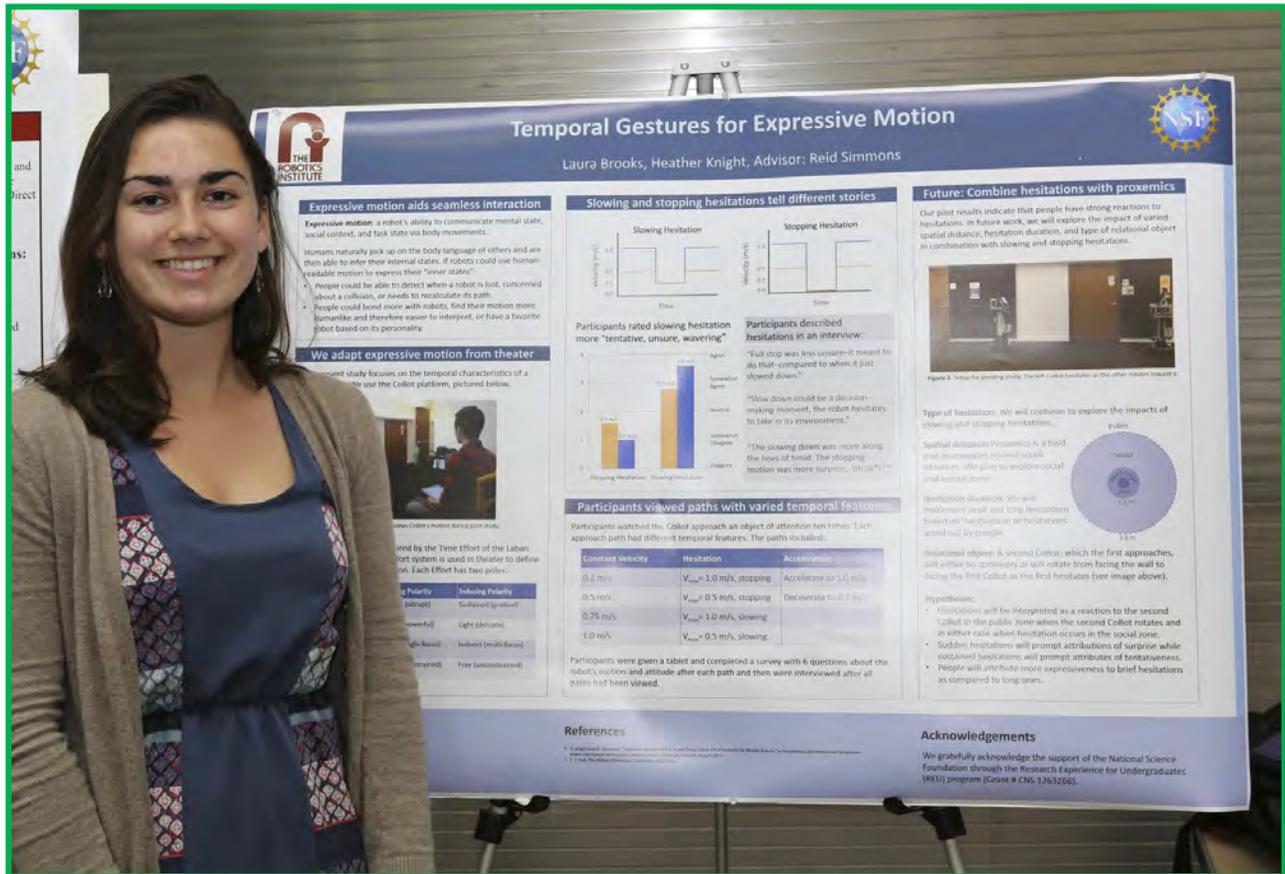
- [1] J. Smisek, M. Jancosek and T. Pajdla. 3D with Kinect. CMP, Dept. of Cybernetics, FEE, Czech Technical University in Prague, 2011.
- [2] N. Burrus. Kinect calibration. <http://nicolas.burrus.name/index.php/Research/KinectCalibration>. 1, 2
- [3] P. Kovesi. Ransac Code. School of Computer Science and Software Engineering, the University of

Western Australia, 2003 – 2005.

- [4] Microsoft. Kinect for X-BOX 360. <http://www.xbox.com/en-US/kinect>, 2010. 1

Laura Brooks

RISS 2015



The poster is titled "Temporal Gestures for Expressive Motion" and is presented by Laura Brooks and Heather Knight, advised by Reid Simmons. It is a poster for the National Science Foundation (NSF) program. The poster is divided into several sections:

- Expressive motion aids seamless interaction:** Discusses how expressive motion (robot's ability to communicate mental state, social context, and task state via body movements) can help humans interact with robots. It notes that humans naturally pick up on the body language of others and are then able to infer their internal states. It also mentions that people could find more with robots, find their motion more humanlike and therefore easier to interpret, or have a favorite robot based on its personality.
- We adapt expressive motion from theater:** Mentions a recent study focuses on the temporal characteristics of a new use of the Colbot platform, pictured below.
- Slowing and stopping hesitations tell different stories:** Contains two graphs showing velocity (m/s) vs. time for "Slowing Hesitation" and "Stopping Hesitation". Below the graphs, it states: "Participants rated slowing hesitation more 'tentative, unsure, wavering'" and "Participants described hesitations in an interview." It includes quotes from participants: "Full stop was less unsure—it meant to do that—compared to when it just slowed down." and "Slow down could be a decision-making moment, the robot hesitates to take in its environment." It also notes: "The slowing down was more along the lines of 'slow'. The stopping motion was more surprise, like 'oh my!'"
- Participants viewed paths with varied temporal features:** Lists paths with different temporal features: Constant Velocity, Hesitation, and Acceleration. A table summarizes the paths:

Constant Velocity	Hesitation	Acceleration
0.2 m/s	$V_{max} = 1.0$ m/s, stopping	Accelerate to 1.0 m/s
0.5 m/s	$V_{max} = 0.5$ m/s, stopping	Decelerate to 0.2 m/s
0.75 m/s	$V_{max} = 1.0$ m/s, slowing	
1.0 m/s	$V_{max} = 0.5$ m/s, slowing	
- Future: Combine hesitations with proxemics:** Discusses how future work will explore the impact of varied spatial distance, hesitation duration, and type of relational object in combination with slowing and stopping hesitations. It includes a photo of a robot in a public zone and a diagram of a public zone with a robot and a person.

References and Acknowledgements are also included at the bottom of the poster.

Temporal Gestures for Expressive Motion

Laura Brooks, Heather Knight, and Reid Simmons

Abstract—Humans intuitively pick up on body language and infer the internal states of those around them. If robots could use human-readable motions to express their internal states and attitudes toward goals, humans would interact more seamlessly with them. Our goal is to express humanlike attitudes in non-anthropomorphic robots with limited degrees of freedom. In order to achieve this, we create gestures using the Laban Effort System, a set of principles from theater that define and help reproduce movement. In this experiment, we specifically focus on the Time Effort. In order to get an understanding of how humans read temporal gestures in robots, we conducted a study asking participants to judge the internal state of a CoBot based on its approach path to an attentional goal. A pilot study found that participants had strong reactions to hesitations and noted differences in expression between slowing (sustained) and stopping (sudden) hesitations. A formal study is pending in which we will explore the impact slowing and stopping hesitations in combination with varied hesitation duration, proxemics, type of relational object, and motion of relational object using videos of CoBot posted on Amazon Mechanical Turk. Based on our preliminary results, we expect the types of hesitations to express different attitudes and interact with the other variables introduced.

Index Terms—expressive motion, Laban Effort System, temporal gestures.

I. INTRODUCTION

IT is becoming more common for people to interact with robots in everyday environments, therefore it is important to facilitate seamless interactions so that people will be able to understand robots' states as intuitively as they understand human states.

Our research focuses on generating expressive motion for robots. We define expressive motion as a robot's ability to communicate mental state, social context, and task state via body movements [1]. Expressive motion is one way of helping humans become aware of the internal state of a robot.

Humans use expressive motion naturally, often involuntarily. Humans also pick up on expressive motion of others and can infer the internal states of those around them. For robots, expressive motion is important to express humanlike attitudes without human motion. If robots could use human-readable motion to express their internal states, both social and functional interaction would benefit. People would be able to

detect when a robot is lost, is concerned about a collision, or needs to recalculate its path. Expressive motion can also improve social interaction with robots. If robots could be consistently expressive enough to have personality, people may bond with robots based on their personality or choose a favorite robot based on its personality.

The Laban Effort System is used in dance and theater to reproduce motion [2]; we use it to create gestures for robots. In non-anthropomorphic robots, it is not possible to simply copy human movements to the robot. The Laban System helps us generate gestures that express human-like attitudes rather than copy human-like movements using human-like robots.

TABLE I
LABAN EFFORT SYSTEM

<i>Time Effort</i>	<i>Fighting Polarity</i>	<i>Inducing Polarity</i>
Time: attitude toward time	Sudden (abrupt)	Sustained (gradual)
Weight: force or apparent inertia	Strong (powerful)	Light (delicate)
Space: attitude toward target	Direct (single-focus)	Indirect (multi-focus)
Flow: sense of restriction	Bound (constrained)	Free (unconstrained)

To create and evaluate these gestures, we use the CoBot robots, 1.3m tall robots with omnidirectional bases which are used in the Gates-Hillman Center at Carnegie Mellon to complete tasks such as delivering messages or showing a person to a location in the building. This platform helps us to isolate the specific Effort we are studying because the CoBots do not have other expressive features.

The present study focuses on the Time Effort. We referred to the Laban Effort system to create velocity settings along a scale from sudden (abrupt) to sustained (gradual), as well as accelerating, decelerating, and hesitation gestures. Both stopping (sudden) and slowing (sustained) hesitations were created. See Table I for complete descriptions of the Laban Efforts.

In a pilot study, participants reacted strongly to hesitations. Both survey and interview questions revealed that participants found the robot to express different things based on the type of hesitation and these findings motivate further work to explore

Submitted for review August, 2015.

We gratefully acknowledge the support of the National Science Foundation through the Research Experience for Undergraduates (REU) program (Grant # CNS 1263266). RISS is an NSF CISE REU site.

L. Brooks, Tufts University, Medford, MA 02155 USA (e-mail: itslaurabrooks@gmail.com).

H. Knight, Robotics Institute at Carnegie Mellon University, Pittsburgh, PA 15289 USA (e-mail: heatherbot@cmu.edu).

R. Simmons, Robotics Institute at Carnegie Mellon University, Pittsburgh, PA 15289 USA (e-mail: reids@cs.cmu.edu).

exactly what kinds of hesitations express specific attitudes.

In a pending video study, we manipulate type of hesitation as well as hesitation duration, spatial distance of relation to goal, type of attentional goal (human or other CoBot), and movement of attentional goal.

II. RELATED WORK

A growing body of research links movement with the ability to express emotions and intentions. As early as 1944, Heider and Simmel showed that people have a tendency to attribute social and emotional qualities to the most basic forms given patterns of motion [3]. In fact, previous work shows that people find gestures in inanimate objects to be expressive. Ju and Takayama demonstrated that people found automatic doors with only one physical degree of freedom to express different attitudes depending on their trajectory [4].

Other work suggests that across different platforms and embodiments of motion trajectories, expression remains the same [5]. It follows that creating low-level gestures with CoBot will help construct a theoretical basis for more complex gestures for many different types of robots.

We hope to use the Laban Effort System to understand how to create expressive motion from the most basic elements of robot movement. The Laban Effort System has been shown to aid in computationally describing motion [1]. While research exists which applies the Laban Effort System to anthropomorphic robots [6] [7], limited research has applied it to non-anthropomorphic robots.

The Laban Effort System has been successfully applied to a flying mobile robot with consistent ratings of valence and arousal based on manipulations of the Efforts [8]. We plan to focus only on the Time Effort in order to isolate and fully understand its implications for expression.

Previous work demonstrates that temporal gestures including speed and acceleration play an important role in perceived attitude of robots [5] [8] [9] [10] [11].

It has been verified that hesitations can be recognized in a robotic arm and likely communicate the same nonverbal message as human hesitations [13]. Hesitations in robotic arms have been found to be readable by humans and in some cases improve the human's rating of the robot's anthropomorphism [14].

While a significant amount of work has been done to understand how hesitations affect collaborative interaction with robotic arms [13] [14] [15] [16], we are interested in integrating hesitations into the basic behavior of a mobile robot with limited degrees of freedom. Some work in the early stages of exploring hesitations in mobile non-anthropomorphic robots suggests that robot hesitations while approaching a human allow the human to continue to be more efficient [17].

One study created the expression of forethought in an animated robot by manipulating the timing of motion, which made the robot seem more appealing and approachable to participants [12]. The manipulation of timing is similar to our hesitation gestures.

More work is needed to fully understand the features of robot hesitations are how they are interpreted by humans. For example, it has been hypothesized but not proven that smooth hesitations are more functionally useful than abrupt ones [15]

[16]. We hope to use the Laban Time Effort to gain this understanding and find parameters for controlling expression.

III. PILOT STUDY

A. Participants

Seven participants (4 male, 3 female) volunteered to take part in our pilot study. All were undergraduates in the fields of computer science, robotics, and/or engineering asked to participate by the researchers. There were no other eligibility or exclusion criteria and participants were not compensated.

B. Design

Participants were seated in an atrium in the Gates Center for Computer Science at CMU for the duration of the study. From there, they viewed the CoBot travel in a hallway toward an attentional goal (Fig. 1). We chose to use an inflatable cactus as the attentional goal because it would be easily distinguished from any other possible attentional goal in the hallway.



Fig. 1. A participant observes the CoBot during pilot study.

Participants observed the CoBot complete 10 different approaches, each a straight line path with the same start and end points. Temporal features of each path were varied:

Constant velocity: Four paths were at constant velocities of 0.2 m/s (CoBot's minimum speed), 0.5 m/s, 0.75 m/s, and 1 m/s.

Acceleration: In two paths, velocity increased or decreased. In one path, the robot accelerated from 0.2 m/s to 1 m/s and in another it decelerated from 1 m/s to 0.2 m/s.

Hesitations: Finally, four hesitation paths were included. Two types of hesitations were included: sudden (stopping) hesitations and sustained (slowing) hesitations. Sudden hesitations involve the robot travelling at a constant velocity, coming to a full stop, then continuing at its original velocity. Sustained hesitations involve the robot travelling at a constant velocity, slowing to its minimum speed, then continuing at its original velocity. Each type of hesitation was completed with a starting velocity of each 0.5 m/s and 1 m/s. In all cases, hesitations occurred at the same location in the middle of the path. Fig. 2 shows the velocity over time of hesitation gestures.

TABLE II
TEMPORAL APPROACH PATHS

Constant Velocity	Acceleration	Hesitation
0.2 m/s	accelerate to 1.0m/s	$V_{\max} = 0.5\text{m/s}$, sudden
0.5 m/s	decelerate to 0.2m/s	$V_{\max} = 1.0\text{m/s}$, sudden
0.75 m/s		$V_{\max} = 0.5\text{m/s}$, sustained
1.0 m/s		$V_{\max} = 1.0\text{m/s}$, sustained

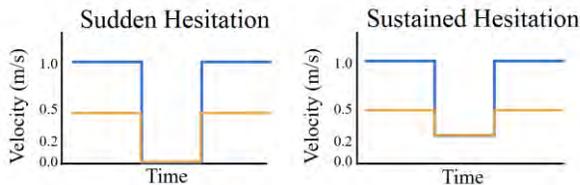


Fig. 2. Velocity over time is shown for hesitation gestures. The blue lines represent paths with starting velocities of 1 m/s while the orange lines represent paths with starting velocities of 0.5 m/s.

Participants were given a tablet at the start of the experiment and after viewing each approach path, they completed a 6 item questionnaire. All items were statements and participants rated how much they agreed with each on a 5 point Likert scale. The purpose of the first four questions was to determine the legibility of the Laban Efforts of Time and Weight. Weight was included because acceleration introduces possible force behind movements. The final two items were meant to measure expressiveness. They read “The robot’s motion is tentative, unsure, wavering” and “The robot’s motion is urgent, overexcitable, hasty.” These final two items were added to the survey after the first two participants had completed the study, therefore they were only answered by 5 participants (3 male, 2 female).

An interview session followed the viewing of all approach paths. We spoke with participants for about 10 minutes in order to understand any of their perceptions that may not have been captured in the questionnaire. We began by asking a very open-ended question about any motions or approaches that may have stood out to them. This was followed by conversation and questions about which paths looked most humanlike or artificial, when the robot appeared to be paying attention or not paying attention to the goal, when it seemed hurried or not hurried, whether they noticed the difference between hesitations, and any storytelling or narrative they imagined to explain or describe the robot’s motion.

C. Results

Means and standard errors for all questionnaire items can be seen in Fig. 4.

1) Constant velocities:

Survey results showed a trend in which participants tended to perceive the maximum velocity as sudden and all slower

velocities as sustained, see Fig. 3. In the interview session, many participants mentioned this.

Participant 6 summarized the feeling that most participants expressed in some way about velocities slower than 1 m/s by sharing their thought, “Why would it go slower than max speed?”

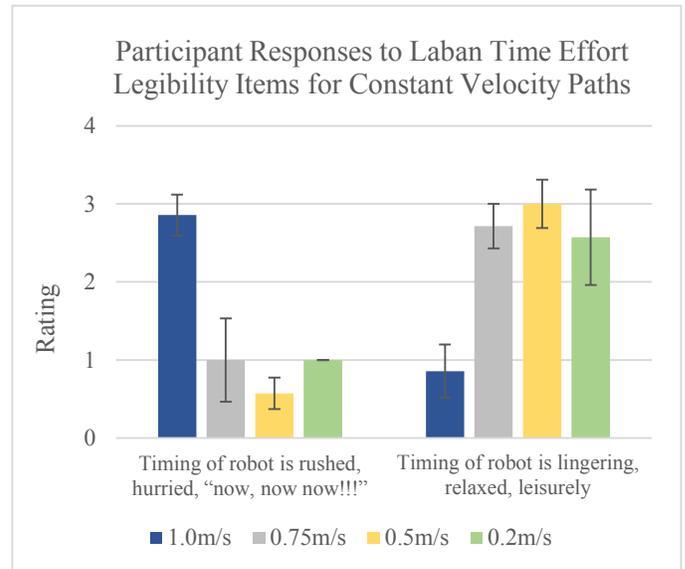


Fig. 3. This chart shows mean participant ratings of survey items for legibility of the Laban Time Effort. A score of 0 corresponds to “disagree,” 2 corresponds to “neutral,” and 4 corresponds to “agree.” Error bars represent standard error.

2) Accelerations:

Participants’ ratings showed a trend that accelerations mapped more to the strong pole of the Weight Effort while decelerations mapped more to the light pole of the Weight Effort, however these associations were not very strong. Multiple participants commented that these paths would express something more clearly if there was a segment of constant velocity and then acceleration or deceleration toward the goal. Several participants commented that the decelerating path made the robot seem like it knew what its goal was and that it was paying attention to it. Others said that it was becoming less sure of its goal as it slowed down. Some participants said that acceleration made the robot look hurried, while one participant said the opposite because it started out so slowly. An accelerating segment at the end of a constant velocity path may elicit more consistent attributions of expression.

3) Hesitations:

Participants did find hesitations to be quite expressive. In particular, there was a notable difference between the perceived attitude of the robot between sudden and sustained hesitation. Generally, the sudden hesitation was associated with attributions of surprise or excitement, while the sustained hesitation was associated with attributions of uncertainty. Fig. 5 shows that participants rated the sustained hesitations more “tentative, unsure, wavering” than sudden hesitations. The

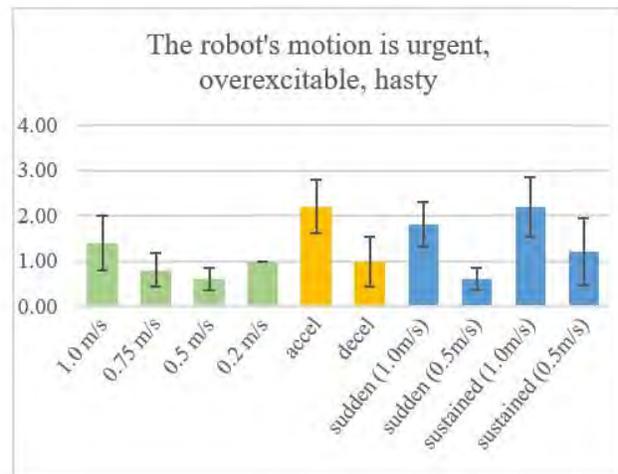
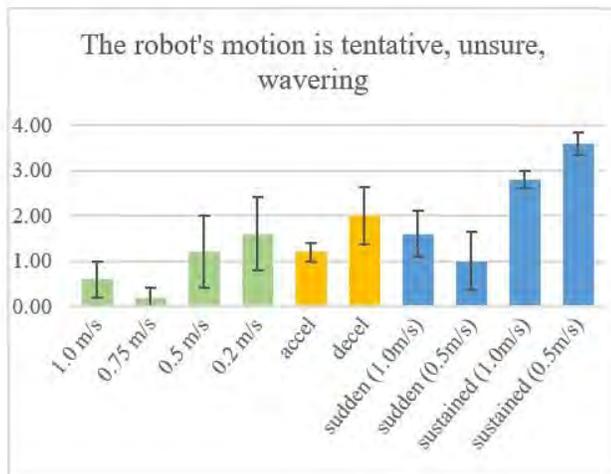
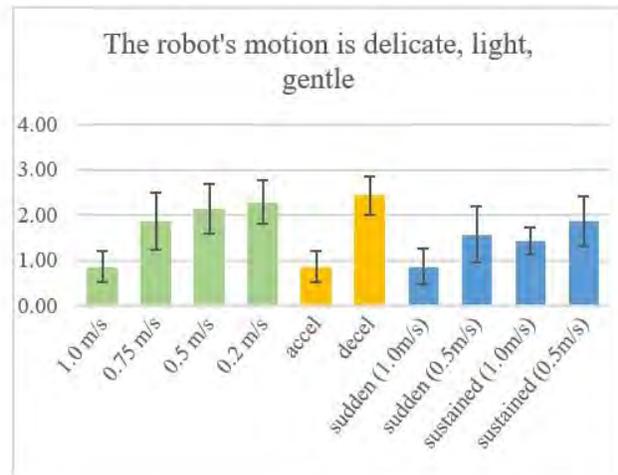
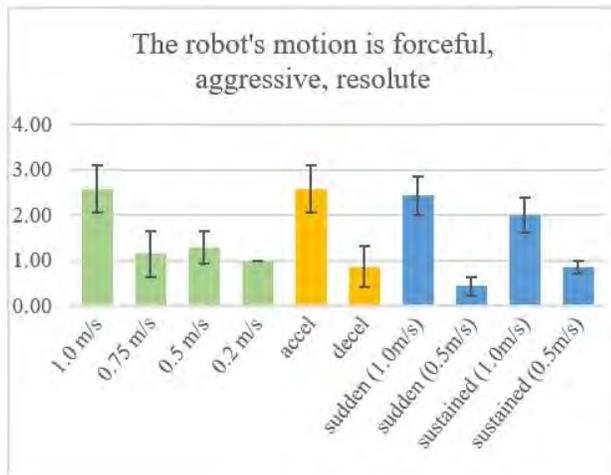
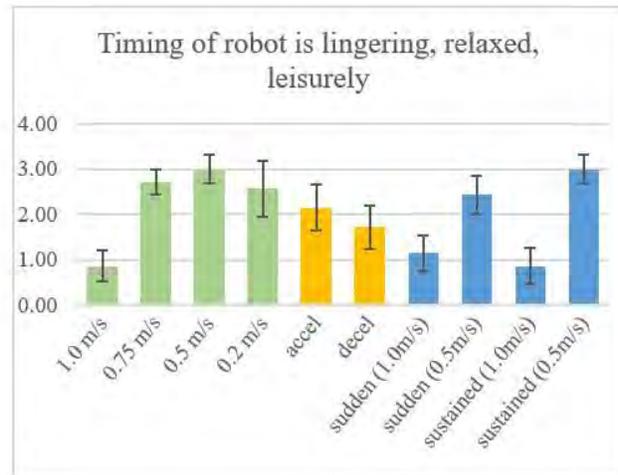
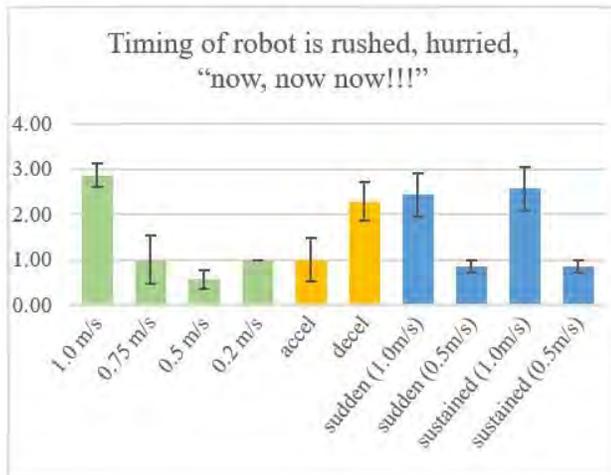


Fig. 4. This set of charts shows mean participant ratings of survey items for each question. A score of 0 corresponds to "disagree," 2 corresponds to "neutral," and 4 corresponds to "agree." Error bars represent standard error. The first four bars in green represent constant velocity paths, while the next two bars in yellow represent accelerations and the final four blue bars represent hesitations.

following are several quotes from participants describing the hesitations in an interview:

- “Full stop was less unsure—it meant to do that—compared to when it just slowed down.”
- “Slow down could be a decision-making moment, the robot hesitates to take in its environment.”
- “The slowing down was more along the lines of timid. The stopping motion was more surprise, ‘oh sh*t!’”

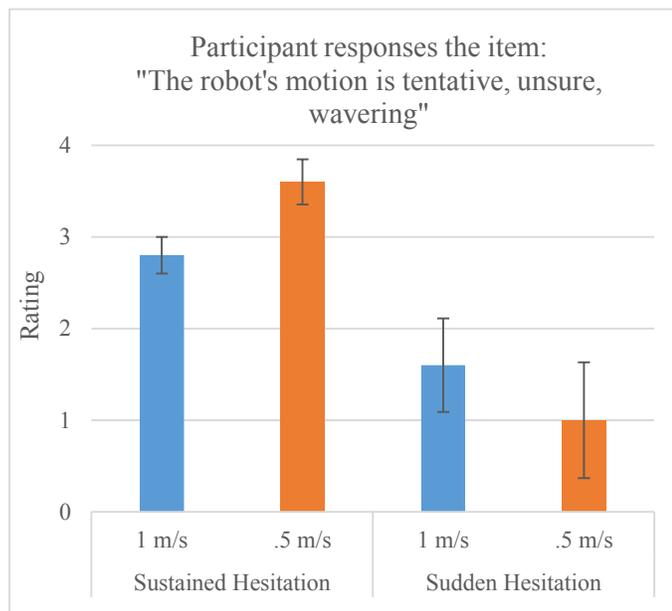


Fig. 5. This chart shows participant ratings on one survey item of different hesitations. A score of 0 corresponds to “disagree,” 2 corresponds to “neutral,” and 4 corresponds to “agree.” Error bars represent standard error. This data is a subset of the data represented in Fig. 4

IV. VIDEO STUDY

Given the strong reactions that participants had to hesitation gestures, we plan to further our research by exploring the impact of various features of hesitations.

A. Variables

Type of hesitation: We will continue to explore the impacts of sudden and sustained hesitations.

Spatial distance: Proxemics is a field that enumerates normal social distances. We will manipulate the location of the hesitation to be in either the personal zone of the relational object (1.2 meters) or public zone or the relational object (4 meters) [18].

Hesitation duration: Based on hesitations acted out by people, we have implemented brief (1.5 s) and long (3 s) hesitations.

Type of relational object: There will be two conditions for attentional goal in this study. The attentional goal will be either a person or a second CoBot.

Movement of relational object: The relational object (human or CoBot) will either remain stationary or will rotate from facing the wall to facing the moving CoBot as it hesitates (see Fig. 6).

B. Procedure

Videos of all combinations of these variables will be created in the same hallway in the Gates Center as the pilot study took place. These videos will be posted to Amazon Mechanical Turk with survey questions. The survey questions will include items that assess the legibility of the Laban Efforts, storytelling that occurred based on robot movement, and the relationship between the moving CoBot and its attentional goal.



Fig. 6. Setup for future study. The CoBot on the left approaches the other CoBot, which rotates toward the first as it hesitates.

C. Hypotheses

1. Hesitations will be interpreted as a reaction to relational object in the public zone when the relational object rotates and in either case when hesitation occurs in the personal zone.
2. Sudden hesitations will prompt attributions of surprise while sustained hesitations will prompt attributions of tentativeness.
3. People will attribute more expressiveness to brief hesitations as compared to long ones.

V. FUTURE WORK

Based on the results of our pending video study, we plan to integrate hesitation gestures into the behavior of the CoBots. We hope to make it easier for people to pick up on a CoBot’s inner state as it navigates hallways in the Gates-Hillman Center. For example, we plan to make the CoBot express when it is in a rush or confused so that people will know whether it is a good time to give CoBot another task or if it needs re-localization or other assistance.

VI. CONCLUSIONS

The results of our pilot study help us to get a sense of how people interpret temporal gestures in robots. We have found promising methods for expressing a non-anthropomorphic robot’s internal state.

Our pilot results suggest that up to 1 m/s, people do not find the robot to be rushed when moving at a constant velocity. Participants did not understand why the robot would ever travel below its maximum speed.

Participants found accelerating/decelerating paths to be a bit confusing because there was no section of constant velocity before accelerations. Some consistency in ratings showed a trend for decelerations to be perceived as light, while accelerations were perceived as strong.

Hesitations elicited interesting responses from participants. Depending on the type of hesitation, the perceived expression of the robot changed. Sudden hesitations were associated with surprise or excitement while sustained hesitations were associated with uncertainty.

Understanding the effects of creating gestures with the Laban Time Effort is one step in being able to generate fully controllable expressive motion for robots. This could aid in creating seamless interaction between humans and robots, as humans would be able to quickly pick up on a robot's internal states—whether it is rushed, lost, surprised, in need of assistance, etc. Generating perceivable attitudes for robots may also help create personalities for robots. If people can relate to a robot via its personality, they may work better with the robot or be less likely to become frustrated with it.

In order to continue improving expressive motion for robots, this work is being continued. Fully understanding the effects of temporal gestures, then layering them with other Laban features will help generate expressive motion for robots of all kinds.

ACKNOWLEDGMENT

We would like to thank everyone who made the Robotics Institute Summer Scholars program possible, especially Rachel Burcin, and the CoBot team for their support.

REFERENCES

- [1] H. Knight and R. Simmons, "Expressive Motion with X, Y and Theta: Laban Effort Features for Mobile Robots," in *International Symposium Robot and Human Interactive Communication (RO-MAN '14)*, Edinburgh, Scotland, 2014.
- [2] R. Laban. *Modern Educational Dance*. Macdonald & Evans, 1963.
- [3] F. Heider and M. Simmel, "An Experimental Study of Apparent Behavior," *The American Journal of Psychology*, vol. 57, no. 2, pp. 243-259, 1944.
- [4] W. Ju and L. Takayama, "Approachability: how people interpret automatic door movement as gesture," *Internation Journal of Design*, vol. 3, no. 2, 2009.
- [5] M. Saerbeck and C. Bartneck, "Perception of affect elicited by robot motion," in *HRI '10 Proceedings of the 5th ACM/IEEE international conference on Human-robot interaction*, 2010.
- [6] M. Masuda, S. Kato and H. Itoh, "Laban-based motion rendering for emotional expression of human form robots," in *Knowledge Management and Acquisition for Smart Systems and Services, Volume 6232*, Springer Berlin Heidelberg, 2010, pp. 49-60.
- [7] M. Masuda, S. Kato and H. Itoh, "Emotion detection from body motion of human form robot based on laban movement analysis," in *Principles of Practice in Multi-Agent Systems, Volume 5925*, Springer Berlin Heidelberg, 2009, pp. 322-334.
- [8] M. Sharma, D. Hildebrandt, G. Newman, J. E. Young and R. Eskicioglu, "Communicating affect via flight path: exploring use of the laban effort system for designing affective locomotion paths," in *HRI '13 Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*, 2013.
- [9] H. Mizoguchi, T. Sato, K. Takagi, M. Nakao and Y. Hatamura, "Realization of expressive mobile robot," in *Proceedings 1997 IEEE International Conference on Robotics and Automation*, Albuquerque, NM, 1997.
- [10] D. Carton, A. Turwald, D. Wollherr and M. Buss, "Proactively approaching pedestrians with an autonomous mobile robot in urban environments," in *Experimental Robotics, Volume 88*, Springer International Publishing, 2013, pp. 199-214.
- [11] H. Knight and R. Simmons, "Taking candy from a robot: Speed features and candy accessibility predict human response," in *International Conference on Robot and Human Interactive Communication (Ro-Man '15)*, Kobe, Japan, 2015.
- [12] L. Takayama, D. Dooley and W. Ju, "Expressing thought: improving robot readability with animation principles," in *Proceedings of the 6th International Conference on Human Robot Interaction, HRI 2011*, Lausanne, Switzerland, 2011.
- [13] A. Moon, C. Parker, E. Croft and H. Van der Loos, "Did you see it hesitate? - Empirically grounded design of hesitation trajectories for collaborative robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, CA, 2011.
- [14] J. W. Hart, B. Gleeson, M. Pan, A. Moon, K. MacLean and E. Croft, "Gesture, gaze, touch, and hesitation: Timing cues for collaborative work," in *HRI Workshop on Timing in Human-Robot Interaction*, Bielefeld, Germany, 2014.
- [15] A. Moon, C. A. C. Parker, E. A. Croft and H. F. M. Van der Loos, "Design and impact of hesitation gestures during human-robot resource conflicts," *Journal of Human-Robot Interaction*, vol. 2, no. 3, 2013.
- [16] L. Riek, T. Rabinowitch, P. Bremner, A. Pipe, M. Fraser and P. Robinson, "Cooperative gestures: Effective signaling for humanoid robots," in *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Osaka, Japan, 2010.
- [17] C. Dondrup, C. Lichtenhäler and M. Hanheide, "Hesitation signals in human-robot head-on encounters," in *HRI '14 Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, 2014.
- [18] E. T. Hall, *The Hidden Dimension*, Garden City, NY: Doubleday, 1966.

Nai Chen Chang

RISS 2015



Differentiating Singularity and Multiplicity in Web Images

Nai Chen Chang, Xinlei Chen, Abhinav Gupta
Carnegie Mellon University

Abstract—Learning to visually count is a strategy that has been recently proposed and utilized to assist in estimating the number of objects in an image. We propose a new counting framework to enrich object recognition databases. Our goal is to be capable of differentiating between images with single or multiple objects. However, we do not attempt the task of learning object detection and counting the various localizations of these objects. Rather we pursue the phenomenon of subitizing, which is the scenario where humans can instantaneously recognize whether an image contains one or more objects. For this problem, we use a new image dataset composed of web images with different numbered objects. In order to learn the representations of singular and multiple objects, we train a Convolutional Neural Network. We are able to show that our network obtains a 55.79% accuracy in predicting if an image contains one or more objects. We also use this network to predict whether the resulting dataset from an object recognition neural net contains undesired images of multiple object instances.

I. INTRODUCTION

While visual knowledge bases, such as the NEIL database [1], are able to establish relationships to label images from a big scale of visual data from the web, the database labels multiple instances of an object as a single instance instead. These web images with multiple occurrences of the same object do not contribute additional information to a labeled cluster. In an attempt to enrich these visual knowledge bases, we pursue the task of differentiating between singular and multiple instances in web images, a counting problem.

The counting problem is the estimation of the number of instances of an object in an image. In an image with a cluttered background, the most important aspects of the image are the salient objects, those that quickly draw viewers' attentions. One approach is training an object detector and an object localization counter, which is computationally intensive. Thus, we implement a phenomenon known as "subitizing", named by E.L. Kaufman et al [2]. Subitizing the inherent human capability to rapidly and accurately determining 1, 2, 3, or 4 salient objects easily with a quick glance. Beyond 4 objects, it has been shown that human response time increases and accuracy decreases. A quantitative model of real time versus numerosity function has been proposed: the function is separated into two regions [7][10][14]. The first region has a small slope with a 50ms linear increase from 1 to 4 objects, corresponding to subitizing. The second region has a larger slope with approximately 250ms linear increase from 4 to 7 objects, corresponding to counting. With physical enumeration taking a longer response time and having a lower accuracy, our proposal for identifying multiple objects is constrained

to the set of 1, 2, 3, or 4 objects in subitizing. Furthermore, with a steady linear increase between every additional objects, precise estimation of instances is eliminated from our learning algorithm. Instead, we use a simple approach using subitizing to distinguish images with a single object from images with more than one object.

II. RELATED WORKS

Recently there have been a number of methods used to address counting problems. However, many of these approaches need object localization or previous knowledge of the type of object to count. A method of approaching the counting problem is to count objects with visual detection. This method requires a visual object detector that can localize all instances of an object in an image [13]. Once the localizations are completed, counting is simply a matter of the number of localizations. The method is highly dependent on accurate object detection followed by localization. However, object recognition is still a nontrivial task given occluding objects and cluttered backgrounds. Some methods augment detection with assumptions regarding interleaved relationships between objects and instances [8]. Others constrain counting to a specific set of objects such as cells and pedestrians [11][12], where the approaches segment objects, extract the features, and deduce the number of objects with supervised learning. While these methods provide accurate results, the assumptions prevent them from being reliable for more complicated and diverse images, such as those found in Google Image Search.

Additionally, there are existing approaches that specifically deal with salient objects [9]. Salient object detection localizes objects with foreground masks or bounding boxes. However, these methods are often trained for single salient objects and are unreliable as a detector for counting objects. In [3], they attempt to use Convolutional Neural Networks to learn deep features for 1, 2, 3, or 4 salient objects. However, while the approach implements subitizing, the results are not favorable past 1 object. Multiple objects are often confused with neighboring counts (i.e. 2 for 3 and 2 for 1). Thus, to reduce complications, we suggest to be indiscriminate regarding multiple objects. Our interests lie with the simple differentiation between single and multiple objects.

III. TECHNICAL APPROACH

With our goal to utilize the concept of subitizing to learn the difference between single and multiple instances of an object in web images, we implement a method to detect the number

salient objects without using any counting or object detection algorithms.

A. Database

Our image database is shown in Figure 1. In order to introduce a variety of objects in web images, we collected images from a list of 2000 categories. The categories are objects in singular form, such as "car" and "apple". To ensure that images have 2, 3, or 4 instances of objects in the images, the queries are pluralized and augmented with the words "two", "three", and "four". We use Google Image Search to download up to one thousand images from each of the categories. Then, our method iteratively cleans the images to the correct format. In total, we downloaded approximately 2.5 million images to use as our image dataset.



Fig. 1. Examples of images downloaded from Google Image Search.

B. Caffe

Recently there have been numerous successes and progressions made by Convolutional Neural Networks [5][16]. Thus, with the large amount of data, we fine-tune a pre-trained CNN model which is proven to be an effective method. We fine-tune the CNN model from [5] because the model is trained on a variety of images from ImageNet, as eclectic as our downloaded images. The final layer is changed to a total of 2 outputs for a binary classification: 0 for multiple objects and 1 for single object. We use Caffe for fine-tuning the pre-trained model. All images are automatically re-sized to 256 by 256 without using aspect ratios of each individual image. Furthermore, four 227 by 227 crops of each image from its bottom-right, bottom-left, top-left, and top-right areas are used to add to the training data of the model. Learning rate is altered to 0.001, and batch sizes are decreased to 128. Because the model is fine-tuned and not trained from scratch, the max number of training iterations and learning rate step size are decreased by a factor of 10 to 45000 and 10000 respectively.

C. Pipeline

To improve knowledge bases such as NEIL, we create a pipeline that takes in the images in these bases and outputs

whether these images contain single or multiple salient objects. In the case of NEIL, most images are in clusters under categories. We take these clusters and run them with the Alexnet model [5] in order to extract fc7 features. These features are then clustered using k-means, and finally we run the clustered features through our trained model. Thus, we are able to determine if clusters in knowledge bases have images with multiple objects. Further, to supplement knowledge bases, we gather random sub-samples of an image using Selective Search [15] to put through the Alexnet model and extract fc7 features. Doing so informs if an image patch could potentially contain useful single instances of an object. The patch can then be added a cluster.

IV. EXPERIMENTAL RESULTS

The model is trained on downloaded images from Google Image Search, but the entirety of the set is not used. Due to the 3 to 1 ratio for images with many objects to one object, a quarter of images are randomly sub-sampled from each "two", "three", and "four" queries. Thus, the model is trained on a total of approximately 1 million images. For initial testing purposes we randomly select 100 images from the downloaded images, and the model achieves a 71% accuracy. For 1000 randomly sampled images from the downloaded images, the model achieves a 65.8% accuracy. Finally testing on the SOS data set from [3], the model achieves a 55.79% accuracy. All testing returns an accuracy greater than 50%, which is better than chance for binary classification. It is unsurprising the SOS data set lowers the accuracy because the images in the data set have occluding objects and complex backgrounds, creating a difficult subitizing problem.

In Figure 2 and 3, there are the top 10 rankings for each class, 0 and 1, from testing done with 100 images. The highest probability for an image to be classified as 0 (multiple) and 1 (single) are respectively 93% and 84%. The disparity between the probabilities may be related to the fact that most web images are not clean images with simple background and an obvious salient object. Observing Figure 2, one can see that the images have fairly clean backgrounds compared to the more cluttered backgrounds in Figure 3.



Fig. 2. Top 10 probabilities for class 1 (single object).

V. FUTURE WORK

While the pipeline is tested on a few categories, intensive experimentation is still needed. Furthermore, more baseline



Fig. 3. Top 10 probabilities for class 0 (multiple objects).

testing needs to be made to conclude the accuracy of our fine-tuned model. This involves testing with hand labeled images and setting thresholds of the Softmax probability, which determines the class of an image. We hypothesize that an inclusion of background images could potentially assist the model with noisy images.

REFERENCES

- [1] X. Chen, A. Shrivastava, and A. Gupta. *NEIL: Extracting Visual Knowledge from Web Data*. In ICCV, 2013.
- [2] E. L. Kaufman, M. Lord, T. Reese, and K. Volkman. *The discrimination of visual number*. The American journal of psychology, 1949.
- [3] J. Zhang, S. Ma, M. Sameki, S. Sclaroof, M. Betke, Z. Lin, X. Shen, B. Price, and R. Mech. *Salient Object Subitizing*. In CVPR, 2015.
- [4] S. Segui, O. Pujol, and J. Vitria. *Learning to count with deep object features*. In CVPR, 2015.
- [5] A. Krizhevsky, I. Sutskever, and G. E. Hinton. *ImageNet Classification with Deep Convolution Neural Networks*. In NIPS, 2012.
- [6] V. Lempitsky and A. Zisserman. *Learning to count objects in images*. In NIPS, 2010.
- [7] M. T. H. Chi and D. Klahr. *Span and Rate of Apprehension in Children and Adults*. Journal of Experimental Child Psychology, 1975.
- [8] B. Leibe, A. Leonardis, and B. Schiele. *Robust object detection with interleaved categorization and segmentation*. International Journal of Computer Vision, 2008.
- [9] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk. *Frequency-tuned salient region detection*. In CVPR, 2009.
- [10] D. Khair and J. G. Wallace. *Cognitive development: An information processing view*. Hillsdale, NJ: Erlbaum, 1976.
- [11] G. Flaccavento, V. Lempitsky, I. Pope, P. R. Barber, A. Zisserman, J. A. Noble, and B. Vojnovic. *Learning to Count Cells: Applications to Lens-free Imaging of Large Fields*. Microscopic Image Analysis with Applications in Biology, 2011.
- [12] A. B. Chan, Z. S. J. Liang, and M. Vasconcelos. *Privacy Preserving Crowd Monitoring: Counting People without People Models or Tracking*. In Computer Vision and Pattern Recognition, 2008.
- [13] M. Nguyen, L. Torresani, F. de la Torre, and C. Rother. *Weakly supervised discriminative localization and classification: a joint learning process*. In ICCV, 2009.
- [14] R. Oyama, T. Kikuchi, and S. Ichibara. *Span of attention, backward masking, and reaction time*. Perception and Psychophysics, 106-112, 1981.
- [15] R. Girshick, J. Donahue, T. Darrell, and J. Malik. *Rich feature hierarchies for accurate object detection and semantic segmentation*. In CVPR, 2014.
- [16] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. *Overfeat: Integrated recognition, localization and detection using convolutional networks*. In ICLR, 2014.

Felipe G. Galiza

RISS 2015



Analytical Inverse Kinematics of a 6-DOF Mantis Robot Arm and Modeling for Minimum Joints Load

Felipe G. Galiza, David S. Touretzky

The Robotics Institute & Computer Science Department, Carnegie Mellon University
Pittsburgh, PA 15213 USA

Abstract— The inverse kinematics of a 6-DOF robot arm is solved analytically in this paper. The Chiara Mantis robot is used as a real world platform. The methodology consists of using the OpenRAVE ikFast module, which receives a XML format file as input, and generates a C++ code file containing the complete 6-DOF inverse kinematics solution. The analytical solution is successfully obtained through this approach, and among other advantages, the same method can be applied to a variety of different robot configurations. As an addition, a scheme is proposed for classifying the multiple inverse kinematics solutions based on choosing the posture, which enables the Chiara Mantis robot to support its own weight efficiently. In conclusion, obtained results show the robot is able to perform accurate trajectories, and the static joints load is computed for each of the solutions in order to classify the best solution.

Index Terms— Chiara Mantis, IkFast, Inverse Kinematics, Robot

I. INTRODUCTION

THE term inverse kinematics stands for computing the position of the each joint, which place the end-effector at a desired position. The Chiara Mantis is a robot based on the praying mantis insect, it can use its front legs to both walk and handle objects. Therefore computing the joint angles, which enables the robot to perform accurate trajectories is unavoidable.

Generally, the solution for the 6-DOF inverse kinematics problem is obtained by numerical methods as the Jacobian Transpose [3], the Cyclic Coordinate Descent (CCD) algorithm [4], Triangulation [5]. Nonetheless, the numerical methods have their limitations as it converges just to one answer out of a bigger range of possible solutions. Previous research shows the analytical solution was determined for some commercial highly articulated robots as the NAO humanoid robot [6].

Felipe G. Galiza Author was a Summer Scholar sponsored by the Brazilian Scientific Mobility Program (BSMP) at the Robotics Institute & Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213 USA. He is now an undergraduate student at the University Center of FEI, São Bernardo do Campo, São Paulo, 09850-901 Brazil. (fgoncalvesgaliza@miners.utep.edu).

Dr. David S. Touretzky Author is a Research Professor in the Computer Science Department and the Center for the Neural Basis of Cognition at Carnegie Mellon University, Pittsburgh, PA 15213 USA. (dst@cs.cmu.edu)

However, the analytical inverse kinematics solution for robots with many degrees of freedom is extraordinarily time consuming, mainly because several trigonometric substitutions are needed. Another drawback is that these trigonometric substitutions are specific to one robot configuration.

The complete inverse kinematics solution of a highly articulated robot leads to a set of multiple answers, therefore choosing the most appropriate solution also becomes a problem to be solved.

Determining a proper solution depends on many factors, starting with the purpose of the movement performed by the end-effector. Thinking about the task of walking, the robot has to be in equilibrium while rising and landing its legs when performing a step, then one feasible strategy can be choosing the solution, which requires the minimum joints angle rotation in comparison to the previous posture.

A more appropriate strategy for controlling equilibrium is checking if the Center of Mass of the robot is inside the Support Polygon [7], and computing the Zero Moment Point (ZMP) [8] to ensure the robot is statically and dynamically stable.

Hardware limitations also have to be considered when solving the inverse kinematics. Servomotors are commonly used in robotics applications, and they have torque limitations. Consequently, the robot cannot be in a posture the static torque required to each servomotor overshoots the manufacturer specifications.

The goal of this work is to describe the application of the OpenRAVE ikFast module [1], developed by Rosen Diankov, which analytically solves inverse kinematics equations for a variety of robot configurations and generates optimized C++ code files.

Furthermore, we also discuss about a method for classifying the multiple inverse kinematics solutions based on choosing the posture, which enables the Chiara Mantis robot to support its own weight efficiently.



Fig. 1. a) The praying Mantis insect. b) The Chiara Mantis robot.

II. METHODOLOGY

The method consists of creating an OpenRAVE ikFast model to describe the relative position of each link and joint of the robot. This is done by writing a XML file with the information about the translation and rotation relative to each component.

The next step is to run a command at the Linux Ubuntu terminal, which generates a c++ code file with the complete inverse kinematics solution for the specified robot configuration.

In part B, an approach for computing the joints static torque is shown, and it can be used as a parameter for choosing the most appropriate inverse kinematics solution.

A. The OpenRAVE ikFast robot model

The OpenRAVE ikFast module requires as an input a XML file format, containing information about the robot configuration, which can be represented by using the Denavit-Hartenberg convention, also called D-H parameters [9].

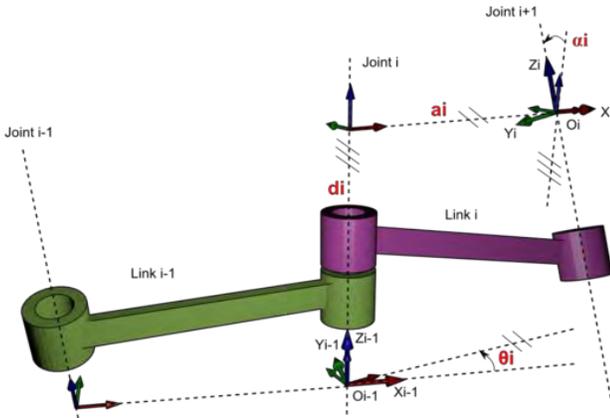


Fig. 2. The Denavit-Hartenberg Convention.

$$A_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \cdot \cos(\alpha_i) & \sin(\theta_i) \cdot \sin(\alpha_i) & a_i \cdot \cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \cdot \cos(\alpha_i) & -\cos(\theta_i) \cdot \sin(\alpha_i) & a_i \cdot \sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

After defining the D-H parameters, each transformation matrix A_i can be used to build the robot model. An example of how to write the XML file is shown below:

```

Robot name="RobotName">
  <KinBody>
    <Body name="BaseFrame" type="dynamic">
      <Translation>0.0 0.0 0.0</Translation>
      <RotationMat>1 0 0 0 1 0 0 0 1</RotationMat>
    </Body>
    <Body name="FirstLinkFrame" type="dynamic">
      <offsetfrom>BaseFrame</offsetfrom>
      <Translation> Ai(3,1) Ai(3,2) Ai(3,3) </Translation>
      <RotationMat> Ai(0,0) Ai(0,1) Ai(0,2) Ai(1,0) Ai(1,1) Ai(1,2)
      Ai(2,0) Ai(2,1) Ai(2,2) </RotationMat>
    </Body>
    <!--
    ... Repeat the same structure for all robot links
    -->
    <Joint circular="true" name="FirstJointFrame"
    type="hinge">
      <Body>FirstLinkFrame</Body>
      <Body>SecondLinkFrame</Body>
      <offsetfrom>FirstLinkFrame</offsetfrom>
      <axis>0 0 1</axis>
      <limits>-180 180</limits>
      <resolution>1</resolution>
    </Joint>
    <!--
    ... Repeat the same structure for all robot joints
    -->
    <Manipulator name="ManipulatorName">
      <effector>LastLinkFrame</effector> <!-- last link -->
      <base>BaseFrame</base> <!-- base link-->
    </Manipulator>
  </Robot>

```

After defining the model and writing the XML file, the OpenRAVE ikFast module is ready to be used. An Linux Ubuntu terminal command example, executed at the same directory where the "robotfilename.xml" is located, is show below.

```

python `openrave-config --python-dir`/openravepy/_openravepy/_ikfast.py --
robot=robotfilename.xml --baselink=0 --eelink=7 --
savefile=filename.cpp

```

The “filename.cpp” will be generated. The file has a function named Computelk that can be used for any application which requires solving the inverse kinematics for the specified robot configuration.

B. Modeling for Minimum Joints Static Torque

The solution of a 6-DOF analytical inverse kinematics problem leads to a set of multiple robot postures. In this case the target is constrained to maintain the foot frame perpendicular to the floor. If we make an analogy with a common robot manipulator, the foot frame is being considered as the base frame. The thorax-head assembly is treated as the end-effector, and its Center of Mass is the point at which forces are applied.

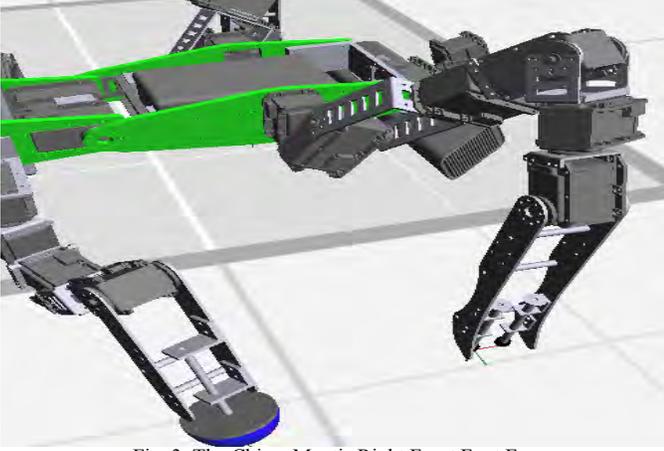


Fig. 3. The Chiara Mantis Right Front Foot Frame.

The movement of the Chiara Mantis joints can be represented by the rotation matrix R_i , which represents a rotation about the local joint z-axis by an angle q_i .

$$R_i = \begin{bmatrix} \cos(q_i) & -\sin(q_i) & 0 & 0 \\ \sin(q_i) & \cos(q_i) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

The transformation matrix from the 6th joint frame to the foot frame is given by T_{foot} . From the 1st joint to the thorax-plus-head Center of Mass frame, the transformation matrix is given by T_{COM} . Therefore the full transformation is represented by $T_{\text{COM}}^{\text{foot}}$.

$$T_{\text{COM}}^{\text{foot}} = [(\prod_{i=1}^6 A_i \cdot R_i) \cdot T_{\text{foot}}]^{-1} \cdot T_{\text{COM}} \quad (3)$$

The Jacobian matrix can be obtained from $T_{\text{COM}}^{\text{foot}}$, by using the following definition:

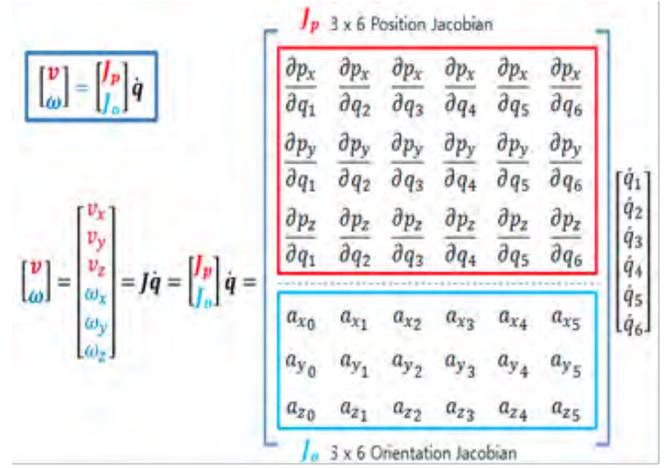


Fig. 4. The Jacobian matrix deduction.

TABLE I
D-H PARAMETERS FOR THE CHIARA MANTIS RIGHT FRONT LEG

Link	d_i	θ_i	α_i	a_i
Base	-25.394	0	90	420.0337
1	-121.65	0	-180	129.09
2	0	0	90	29.0373
3	0	-90	-90	0
4	170.4535	90	90	0
5	0	-90	-90	0
6	134.4633	-90	-90	0
Dummy	0	0	90	152.5614
Foot	-206.5242	9	144.626	0

Linear measurement units are in mm, and angular are in degrees.

Finally, we can define the torque required by each joint to keep the robot in static equilibrium as the vector τ .

$$\vec{\tau} = J^T \cdot \vec{F} \quad (4)$$

Where:

$$\tau = [\tau_1 \tau_2 \tau_3 \tau_4 \tau_5 \tau_6]^T \quad \text{and} \quad F = [F_x F_y F_z T_x T_y T_z]^T.$$

III. RESULTS

At this section we introduce the results of the experiments done by the Chiara Mantis robot simulation model, the software package Tekkotsu [11] and the Mirage simulation environment. Section A shows an experiment with inverse kinematics and section B shows focus on joint static torque analysis.

As stated before in the methodology section, all we need in order to compute the inverse kinematics is the D-H parameters of the robot. The D-H parameters of the Chiara Mantis Right Front Leg are used as the input for the experiments in this section.

A. ikFast Analytical Inverse Kinematics Solution Analysis

The OpenRave IkFast module was applied at the Chiara Mantis 6-DOF front legs. A sample of several points forming a cylinder was used as a trajectory the leg should be able to

follow. At the end of the experiment, the OpenRAVE IkFast function returned from four to eight solutions for each trajectory point. On each iteration the first solution was taken, and the angle vector $q = [q_1 \ q_2 \ q_3 \ q_4 \ q_5 \ q_6]$ was updated at the forward kinematics equation. The length representing the foot frame

at four of the six leg joints.

ACKNOWLEDGMENT

Felipe G. Galiza Author thanks all people who contributed for his experience as a Robotics Institute Summer Scholar, especially Dr. David S. Touretzky for advising him during all course of this project.

REFERENCES

- [1] Rosen Diankov. The analytical Inverse Kinematics Generator. Available: <http://openrave.programmingvision.com/en/main/openravepy/databases.inversekinematics.html>.
- [2] Robert L. Williams. Robot Mechanics. NotesBook Supplement for ME 4290/5290 Mechanics and Control of Robotic Manipulators. Available: <http://www.ohio.edu/people/williar4/html/PDF/Supplement4290.pdf>
- [3] Samuel R. Buss and Jin-Su Kim. "Selectively Damped Least Squares for Inverse Kinematics." In Journal of Graphics Tools, vol. 10, no. 3 (2005) 37-49.
- [4] Jeff Lander. "Oh My God, I Inverted Kine!." In Game Developer. September 1998.
- [5] R. Müller-Cajar, R. Mukundan, "Triangulation: A New Algorithm for Inverse Kinematics", Proceedings of Image and Vision Computing New Zealand 2007, pp. 181-186, Hamilton, New Zealand, December 2007.
- [6] N. Kofinas, "Forward and inverse kinematics for the NAO humanoid robot," Diploma Thesis, Technical University of Crete, Greece, 2012, available at: www.intelligence.tuc.gr/lib/downloadfile.php?id=430.
- [7] C. Queiroz, N. Gonçalves, P. Menezes, A Study on Static Gaits for a Four Legged Robot, International Conference CONTROL/2000, 2000, Cambridge, UK
- [8] M. Vukobratovic, B. Borovac, Zero Moment Point – Thirty five years of its life, International Journal of Humanoid Robotics vol. 1, pp. 157-173, 2004
- [9] Denavit, Jacques; Hartenberg, Richard Scheunemann (1955). "A kinematic notation for lower-pair mechanisms based on matrices". Trans ASME J. Appl. Mech 23: 215-221.
- [10] M. B. Leahy Jr, L. M. Nugent, G. N Saridis and K. P. Valavanis, "Efficient puma manipulator jacobian calculation and inversion." Journal of Robotics Systems, 4(4), 185-197. 1987.
- [11] Ethan Tira-Thompson, David S. Touretzky. "Tekkotsu: A Rapid Development Framework for Robotics". Carnegie Mellon University, Robotics Institute. May 2004.

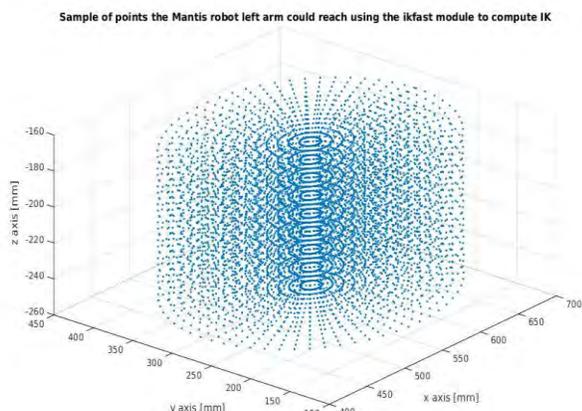


Fig. 5. Inverse Kinematics Experiment

positions after each forward kinematics iteration is shown in Fig. 5.

B. Joints Static Torque Analysis

In this experiment, the target point used for the robot foot frame was constrained to be perpendicular to the floor. The force input vector was composed just by the gravitational force acting at the thorax-plus-head COM. For each solution, the results of the computed absolute static torque at each joint are shown below.

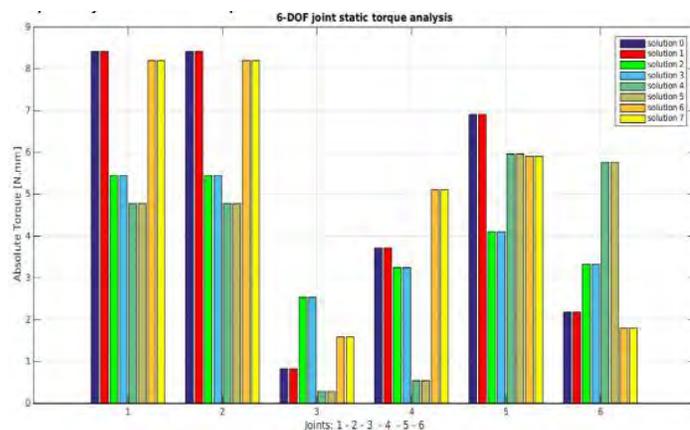


Fig. 5. Joints Static Torque Analysis.

IV. DISCUSSION

The results show the robot could perform accurate trajectories using the OpenRAVE ikFast module, therefore it is a valid approach to solve the 6-DOF Inverse Kinematics problem. In the second experiment, solutions 4 and 5 can be selected as the best solutions since they require the least torque

Jonathan Joo

RISS 2015



Waypoint Sequencing for Planetary Rovers with Significant Time-Dependent Energy Constraints

Paper ID: 2206

Content Areas: HSO: Heuristic Search,
ROB: Motion and Path Planning,
PS: Planning (General/Other)

Abstract

Solar-powered robots will be the first to discover and characterize ices that exist at the poles of some moons and planets. These distinctive regions have extensive, grazing, time-varying shadows that raise significant time and energy constraints for solar-powered robots. In order to maximize the science-value of missions in such environments, rovers must plan to visit as many science targets as possible while taking into consideration the limitations imposed by time-varying shadows. This paper presents two approximation algorithms for selecting and sequencing science waypoints in this environment: a genetic algorithm and a greedy algorithm. Both greatly outperform a brute force approach in calculation time. Furthermore, for large numbers of waypoints, the genetic algorithm shows significant improvements in plan value compared to the greedy algorithm. This research demonstrates that a genetic approach could be utilized to effectively plan future missions for solar-powered rovers in dynamic, shadowed environments.

Introduction

Rover missions to the lunar poles will be shorter and faster-paced than current and previous Mars missions. Due to temperature extremes on the Moon, a lunar rover will not last for years; indeed, missions may even be confined to a two-week lunar sunlight period (George et al. 2012). This contrast in mission lengths necessitates varied approaches to mission planning. Tactical planning for the Mars rovers involves planning out a day or more of operations, waiting for the rover to execute those operations, and then planning future operations based on data the rover returns (Mishkin et al. 2006). Here, strategic planning, which happens on a longer time scale, is decoupled from tactical planning (Barrett et al. 2009). On the Moon however, available communications and abundant solar power enable nearly 24-7 operation with seconds of latency for the mission duration (George et al. 2012; Otten et al. 2015). Tactical decisions can and must be made quickly, but must also carefully consider how a given action might impact subsequent mission goals. For example, stopping for hours to drill in one location could prevent a rover from traveling to another interesting region, as drilling

Copyright © 2015, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

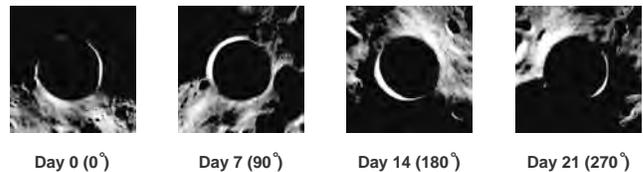


Figure 1: An example of time-varying light conditions on Shackleton Crater, located at the Lunar South Pole.

time coincides with the short time window when the connecting path is illuminated. Thus, the lack of constant sunlight availability can make some regions unreachable with the rover's limited battery capacity. See Figure 1 for an example of time-varying illumination at the lunar poles.

In an ideal scenario, all regions of interest could be visited in a single mission. However, this is clearly not possible with both the time and energy restraints in such a dynamic environment. Thus, planetary rover missions maximize their utility by achieving as many high-value science objectives as possible within their mission lifetimes. These science objectives are associated with spatially distributed waypoints. As the number of waypoints increases, an exact solution becomes impractical (Keller 1989), so the goal of this work is to develop an approximation algorithm that can plan effective waypoint sequences for large numbers of waypoints (~ 100) within a reasonable time frame.

This paper presents and compares two different algorithms – a greedy algorithm, and a genetic algorithm which builds from the greedy solution. These algorithms maximize the science value of visited locations while remaining within energy and time constraints. The planning problem is partitioned into two components. The first is point-to-point planning, which optimizes a path to get from one location to another subject to rover energy constraints. The second is waypoint sequencing, which seeks to determine the optimal subset and sequence of waypoints to visit that maximizes an objective function. Point-to-point planning is used to find the paths and costs between waypoints, which are not known a priori. The genetic and greedy algorithms call the point-to-point planner and use the calculated costs to find high-value sequences of waypoints.

The paper is organized as follows. First, related work and

the exact specification of the waypoint sequencing problem is given. Next, the point-to-point planning algorithm and waypoint sequencing algorithms are presented and explained. Experiments are described and analyzed in the Experiment and Results section. The Conclusions section discusses conclusions and directions for future research.

Related Work

The problem of waypoint sequencing is similar to, but distinct from, several variants of the Orienteering Problem (OP) (Vansteenwegen, Souffriau, and Van Oudheusden 2011). The OP is similar to the well-known Traveling Salesman Problem (TSP), but the agent does not have to visit every goal position, and each goal position provides some reward (Keller 1989). The agent seeks to maximize reward while minimizing path length. Smith addresses the rover mission activity planning problem for Mars rovers (Smith 2004). He describes it as an over-subscription planning problem, where there are too many goals for the time or resources available, and he solves the OP as an intermediate step in planning.

The problem addressed in this research is more similar to variants of the OP: the Time-Dependent Orienteering Problem (TDOP) (Fomin and Lingas 2002), the Orienteering Problem with Time Windows (OPTW) (Kantor and Rosenwein 1992), or the Time-Dependent Orienteering Problem with Time Windows (TD-OPTW) (Garcia et al. 2010). In the TDOP variant, the edge costs between two waypoints are dependent on the time at which the path is begun from one waypoint to another. In the OPTW variant, each waypoint also contains a start time and end time, which specify when the waypoint may be visited. The TD-OPTW variant is a combination of TDOP and OPTW.

The distinction of this work is the the incorporation of rover energy as a non-monotonic resource. With the additional energy constraint, the cost of a path between waypoints not only depends on the time that it is taken, but also the sequence of waypoints and actions taken in the past. The times at which waypoints are available are also controlled by the energy constraints. If, for a period of time, a waypoint requires too much shadowed travel to reach or too much time in shadow to complete an action at the waypoint, energy constraints will prevent travel to that waypoint during that period. Edge costs and waypoints are not known a priori and are sequence dependent, unlike in the OP variants discussed above. However, the sequence affects the edge costs instead of the value gained from waypoints, as in the Sequence Dependent Team Orienteering Problem (SDTOP) introduced by Mennell (Mennell 2009).

Saha et al. investigate the problem of finding a tour of waypoints, (similar to TSP) and finding collision-free paths between waypoints in high-dimensional space (Saha et al. 2006). Like the problem addressed in this work, these paths between waypoints are more expensive to compute than an approximate TSP solution, given the number of waypoints. However, Saha et al. do not deal with time windows or resource constraints.

Researchers have proposed numerous approaches to solving the OP and its variants. Gunawan et. al. (Gunawan, Lau,

and Yuan 2014) pose the TDOP as an integer linear programming model and utilize greedy construction, iterated local search, and variable neighborhood descent as meta-heuristics for a solution. Duque, Lozano, and Medaglia solve the OPTW with the pulse framework, which uses recursive depth-first search with a set of effective pruning methods (Duque, Lozano, and Medaglia 2015), though these pruning approaches require knowing many edge costs that would not otherwise be needed. Wang et. al. (Wang, Golden, and Wasil 2008) and Karbowska et. al. (Karbowska-Chilinska and Zabielski 2014) both utilize a genetic approach to solve the Generalized Orienteering Problem and OPTW, respectively.

Despite the fact that this problem cannot be directly cast as one of the OP variants listed above, some of the algorithms can still be applied. In particular, this paper draws its inspiration from genetic algorithms applied to OPTW due to its ability to start from a population of solutions and iteratively improve upon them. In addition, the implementation of the genetic algorithm does not rely on knowing edge costs and time windows a priori, which is important since they are sequence dependent in this work.

Problem Definition

The problem addressed in this work is waypoint sequencing under time-varying illumination subject to non-monotonic energy constraints. Rover energy dissipates while operating and is recharged during illumination.

Let G be a graph with n vertices (waypoints). P_i denotes the value of a vertex i and T_i represents the amount of time the rover must stay at this vertex before the waypoint can be marked as visited. The rover must remain within the energy constraints during travel between vertices and for the duration of T_i at each vertex. An edge between two vertices contains time, distance, and energy costs. These costs are determined by the point-to-point planner and are a function of starting energy and starting time. Thus, the edge costs and implicit time windows are determined by the energy constraints and preceding actions.

In addition to the graph, start and end vertices are specified (denoted by V_s and V_e). The goal is to find a route through G that begins at V_s and ends at V_e , while maximizing the total value of waypoints visited and staying within the specified energy and total time constraints.

Point-to-Point Planning

Point-to-point planning for solar-powered rovers under time-varying illumination has been addressed both by Tompkins (Tompkins 2005) and more recently Cunningham et al. (Cunningham et al. 2014). Both developed planners that considered energy and time constraints while using deterministic A*-based planning. Otten et al. also addressed long-duration point-to-point path planning on the lunar poles but did not explicitly address energy constraints (Otten et al. 2015). None of these planners addressed the waypoint ordering problem.

This research directly follows the approach of Cunningham et al. to compute time, distance, and energy costs associated with traveling from one waypoint to another at a

given point in time (Cunningham et al. 2014). The planner utilizes an A*-based search algorithm that minimizes time, while ensuring that the rover remains within predefined energy constraints. This planner is used to compute waypoint-to-waypoint travel costs.

In order to represent the dynamic nature of the environment at the lunar poles, the world is discretized into a time series of shadow maps. Each shadow map represents the state of the world for a discrete time interval, τ . A graph is used to encode the world in a representation that a planner can easily interpret. A node of the graph, $N = (x, y, \tau)$, corresponds to a position, (x, y) , at a specific time interval, τ . Edges are created using directed edges in an 8-connected grid. Nodes are connected to other nodes at the same time interval, τ , and at the next time interval, $\tau + 1$. For each pair of connected nodes, there is a set of different velocities that the rover can use that allows different energy and time transitions. Energy and time costs for each edge are calculated using solar power derived from the shadow maps and physics-based rover models.

The point-to-point planner’s A*-based search algorithm minimizes time while keeping all nodes in the path within predefined energy bounds derived from the battery capacity. Energy, E , and time, t , are continuous variables also considered as part of the search state and must stay within predefined constraints. The relevant energy and time values for a state are computed during the search and are not known a priori. Consequently, the planner does not plan directly over the graph nodes. Instead, it dynamically creates a new state $S = (x, y, \tau, t, E)$ every time it adds to the open list. Multiple (x, y, τ, t, E) states correspond to the same (x, y, τ) node in the graph but with different energy and time values. State dominance is used to prevent this method from creating an unbounded number of states.

Energy is a non-monotonic resource cost. It can either increase or decrease over an edge. It has a lower limit that causes states to be pruned and an upper limit (corresponding to maximum battery capacity) after which no more energy can be added.

When the point-to-point planner is asked to plan from a starting state to a waypoint, it plans to a set of goal nodes at that (x, y) location using Euclidean distance divided by maximum speed as the heuristic. Each waypoint has a specified time cost associated with it that models a rover completing its science objective at that location. When the planner opens a state at the goal location, it tests to see whether it can complete its objective at that location and still remain within energy and time constraints before determining that it has completed the search.

Waypoint Sequencing Algorithms

Greedy and genetic algorithms were developed and tested. As a baseline for comparison of results, an exact brute force algorithm was also implemented. These algorithms are outlined below.

Brute Force

The brute force algorithm simply generates all possible sequences of waypoints that have the specified starting and

ending waypoints, and chooses the best sequence. The best sequence is defined as that which has the maximal total sum of the values of waypoints visited. In the case of a tie in maximal values, the sequence with the shortest travel distance is chosen, since minimizing travel distances reduces the risk of rover failure or entrapment. Because the brute force algorithm checks every possible subset and permutation of waypoints, it has a time complexity of $O(n!)$.

Greedy

The greedy algorithm works iteratively, beginning with the specified start waypoint and time. At each step, the greedy algorithm checks if the end can be reached from the current waypoint, and then uses a heuristic function to measure the cost to visit each remaining waypoint. If the end cannot be reached within the time constraint, then the current waypoint is removed and replaced with the final waypoint, ending the sequence. Otherwise, the greedy algorithm chooses the waypoint with the best heuristic value, adds it to a running list of waypoints, and updates the current waypoint to be the new waypoint. The heuristic value function ($H_{i \rightarrow j}$) for traveling from waypoint i to waypoint j is:

$$H_{i \rightarrow j} = \frac{P_j}{C_{i \rightarrow j}} - \frac{\log(P_j)}{P_j + C_{i \rightarrow j}} \quad (1)$$

P_j is the value of waypoint j and $C_{i \rightarrow j}$ is the cost to get from i to j . The heuristic function prioritizes higher value locations. In the worst case, the greedy algorithm examines every waypoint and at every step evaluates the path to each remaining waypoint, resulting in a time complexity of $O(n^2)$.

Genetic

The genetic algorithm functions in a way that models biological processes. First, an initial population of waypoint sequences is generated. Next, a subset of this initial population, called the mating population, is chosen for reproduction. The sequences in this mating population undergo a mating step, generating children that share characteristics of each of the two parent sequences, but perhaps have certain mutations which differentiate them from either parent. These children become the new initial population for the next generation, and the cycle loops until the best sequence in the population is the same as the previous best sequence a total of G_{num} times. This ensures that sequences that have stopped evolving do not take up too much computation time, while more volatile sequences are allowed to run for greater numbers of generations.

Initialization The initialization of the genetic algorithm’s population is accomplished in three steps.

1. Random Generation: ρ_{size} unique sequences are randomly generated. Waypoints are randomly added between the start and end waypoints until there is no longer a valid path through the sequence.
2. Deterministic Greedy Generation: The greedy algorithm (as described above) is used to generate a sequence which is added to the initial population.

3. Probabilistic Greedy Generation: A probabilistic greedy algorithm is also used to generate γ_{size} sequences for the initial population. The probabilistic greedy algorithm is the same as the greedy algorithm described above, but instead of always choosing the best waypoint to add next, it probabilistically chooses among the best three. The probability of choosing the best waypoint is set to 65%, the second-best waypoint at 20%, and the third-best waypoint at 15%. This is done to encourage diversity among generated sequences in the population.

Thus, the initial population consists of a combination of both randomly and greedily generated waypoint sequences, with a total size of $\rho_{size} + \gamma_{size} + 1$.

Fitness Selection In order for evolution to occur, down-selection of the population must happen before mating. A subset of size σ_{size} is chosen from the population to undergo mating, using tournament grouping selection. In tournament grouping selection the sample population is divided into σ_{size} groups of equal size. From each of these groups, the best sequence is added to the mating pool. Tournament grouping selection is shown to result “in individuals after selection that are more diverse” than ordinary tournament selection (Zabieliski et al. 2015). Best fit selection, where the algorithm simply chooses the best σ_{size} sequences, was also evaluated but had a greater tendency to get stuck in local minima.

Evolution The evolution step turns a mating pool into a new generation. It consists of three parts: crossover, mutation, and cleaning.

Crossover: The crossover phase utilizes queen-bee selection and Edge-Recombination Crossover (ERC) (Wang, Golden, and Wasil 2008). In queen-bee selection, the best waypoint sequence in the mating pool undergoes crossover with each of the remaining $\sigma_{size} - 1$ sequences in the mating pool. Edge-Recombination Crossover works as follows:

Suppose two waypoint sequences, I and J , are undergoing crossover. Let A be defined as an adjacency matrix of size $(|I \cup J| - 1) \times 4$. Rows in A correspond to waypoints in the union of I and J , excluding the end waypoint. For each waypoint α in $I \cup J$, the corresponding row in A consists of the preceding and following waypoints of α in I followed by the preceding and following waypoints of α in J . Elements in A are left empty if no such neighbor exists or if the neighbor already exists in that row. For example, suppose $I = \{a, c, d, b, e, f\}$ and $J = \{a, b, e, g, f\}$. Table 1 shows the adjacency matrix for I and J .

Once the adjacency matrix is generated, a sequence is generated. Starting with the start waypoint, V_s , ($V_s = a$ in the example), the algorithm calculates the heuristic value for the paths from V_s to all other waypoints in row V_s , ($H_{a \rightarrow c}$ and $H_{a \rightarrow b}$ in the example above). Then it probabilistically selects the next waypoint, α , from the set of waypoints in row V_s with probability weighted by their heuristic values. α is then added to the current sequence and it is removed from all other entries in A to prevent cycles. Then the algorithm goes to row α and repeats until it reaches the final waypoint or there are no remaining waypoints in the current

Table 1: Example Adjacency Matrix for sequences $I = \{a, c, d, b, e, f\}$ and $J = \{a, b, e, g, f\}$

α	I_-	I_+	J_-	J_+
a	-	c	-	b
b	d	e	a	-
c	a	d	-	-
d	c	b	-	-
e	b	f	-	g
g	-	-	e	f

row. In the latter case, the end waypoint is simply appended onto the existing waypoint sequence. ERC was chosen as the preferred method of crossover, as it offers the advantage of maintaining valid path segments of parents, which are then inherited by children sequences (Wang, Golden, and Wasil 2008). For each pairing of parents, two new children are generated.

Mutation: The mutation step modifies child sequences generated in the crossover step to encourage varied exploration. Each child from the evolution step is copied, and the copy is mutated. The original children stay unchanged. Mutation occurs in three ways:

1. Addition: a random waypoint not currently in the sequence is selected and inserted before the waypoint in the sequence with the lowest Euclidean distance from the selected waypoint.
2. Reversal: a random subsequence within the waypoint sequence is reversed.
3. Swap: two randomly selected waypoints within the sequence switch places in the ordering.

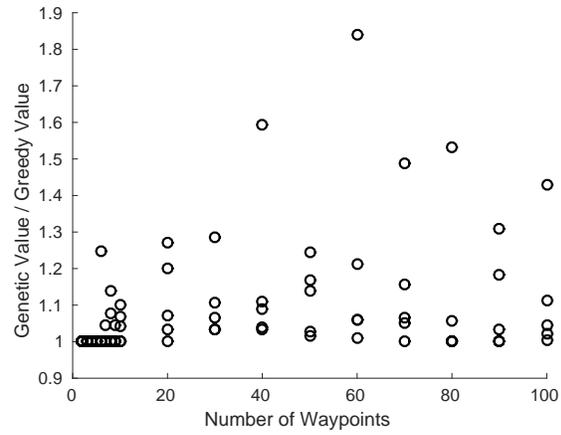
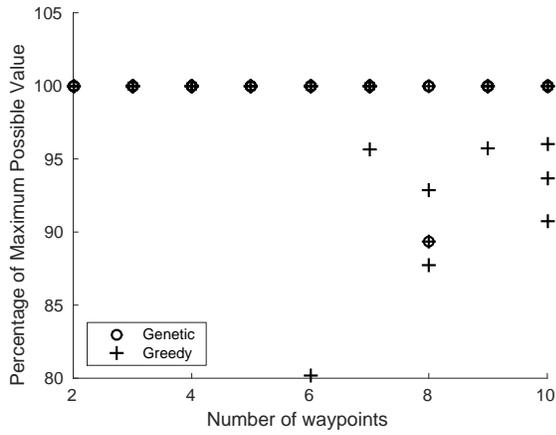
These mutations are all set to occur with a 90% probability. However, one of each pair of children is set to have a 10% chance of a swap mutation rather than a 90% chance, since the swap mutation is more random in nature and is less likely to produce improvements in sequences compared to the other two forms of mutations.

Cleaning: Because some generated children may not be viable paths, the cleaning phase ensures that each waypoint sequence is valid. The cleaning step randomly removes waypoints from a sequence (excluding the beginning and end waypoints) until a viable path exists through the sequence that satisfies all constraints.

After cleaning, the best sequence in the previous generation is also added to the new generation at each iteration to ensure that the quality of the best sequence in the population does not decline. Thus, the final new population size after the entire evolution step is $(4 * (\sigma_{size} - 1)) + 1$.

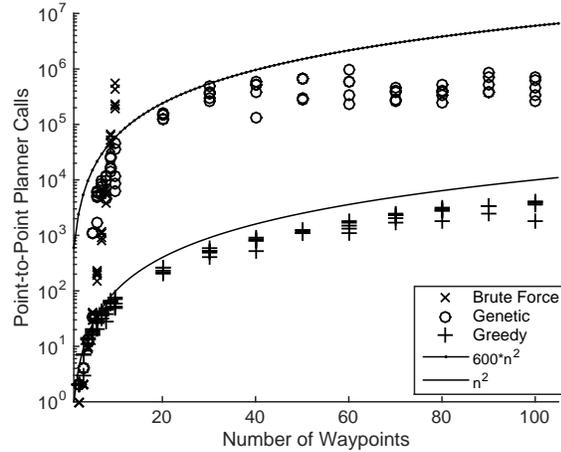
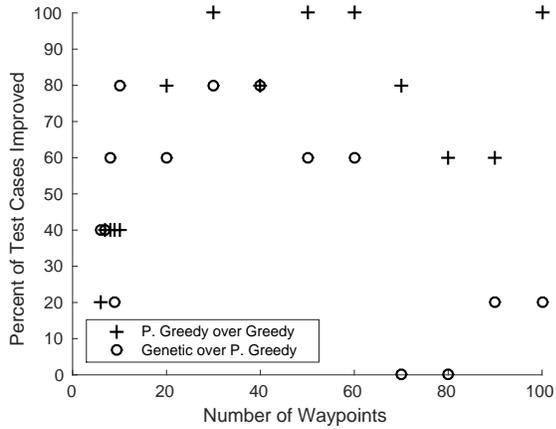
Experiments and Results

Experiments were run to test both the computational efficiency and the quality of paths returned for each waypoint sequencing algorithm. All experiments were run using 35 synthetically-generated shadow maps of Shackleton Crater on the lunar south pole, corresponding to approximately 11



(a) Sequence values from genetic and greedy algorithms as a per-

(b) Sequence values from genetic algorithm divided by the se-



(c) Improvement from greedy to probabilistic greedy and from (d) Total planner calls for each test case and each algorithm as a

function of the number of waypoints.

Figure 2: Results from running each algorithm on five randomly generated cases for each number of waypoints

Earth days. A set of (x, y, τ) node locations and edge costs for the graph used in point-to-point planning were pre-computed and not taken into account in runtime comparisons. All waypoint locations were selected from the set of node locations in this graph.

The number of waypoints and the locations of the waypoints were varied between tests. The following numbers of waypoints were tested: $\{2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$. For each number of waypoints, five different test cases were generated with random waypoint locations and values.

For the graphs containing between 2 and 10 waypoints, inclusive, all three different algorithms (brute, greedy, and genetic) were run on each of the sets of waypoints. However, for sets with more than 10 waypoints, only genetic and greedy were tested due to the $O(n!)$ time complexity of the brute force search. The parameters used for the genetic algorithm are listed in Table 2.

The point-to-point planner and all waypoint sequencing algorithms were implemented in Java. Tests were run re-

Table 2: Genetic Algorithm Parameters

Variable Name and Explanation	Value
ρ_{size} , initial random generation population size	$(n/2)^{1.5}$, max of 200
γ_{size} , initial probabilistic greedy generation population size	σ_{size}
σ_{size} , mating subpopulation size	$n - 2$, max of 20
G_{num} number of generations of no change	10

motely on three M4.2xlarge computers by Amazon Web Services. These computers have 2.4 GHz Intel Xeon® E5-2676 v3 (Haswell) processors, 32 GiB of memory, and 8 vCPUs.

For each waypoint sequencing run, the total number of point-to-point planner calls was used as a measurement of computational complexity. Because the point-to-point planner is called each time a waypoint sequencing algorithm requests an edge-cost, it offers a viable metric to evaluate computational complexity. In addition, for each returned waypoint sequence, the total path distance, total path time, and sum of the value of all visited waypoints is extracted to

evaluate the efficacy of each algorithm. Better waypoint sequences have higher total value, shorter mission times, and shorter travel distances.

Figure 2a shows the total value of waypoint sequences (as a percentage of the total possible value in paths) returned by the greedy and genetic waypoint sequencing algorithms. The total possible value is determined by running the brute force algorithm which guarantees the most optimal result. However, because the brute force algorithm is not practical for more than ~ 10 waypoints, Figure 2b compares only the genetic and greedy algorithms. Here, the graph shows the total value of waypoint sequences returned by the genetic algorithm divided by the total value of waypoint sequences returned by the greedy algorithm. There are five test cases for each number of nodes despite some overlap of data points. For small numbers of waypoints, the path efficiency gained from the greedy to genetic is small. However, as the number of waypoints increases, the improvement of the genetic over greedy is more apparent.

For a given test case, one waypoint sequence is defined as better than another if it has a higher total waypoint value or equal value but takes less time. Figure 2c shows the percentage of test cases improved when comparing the deterministic greedy algorithm to the probabilistic greedy algorithm, used to populate the initial population of the genetic algorithm. It also shows the percentage of test cases improved from the genetic initial population to the genetic final population. Test cases with fewer than six waypoints were not considered. The probabilistic greedy population was better than deterministic greedy in 67% of test cases, and the genetic algorithm improved upon the initial population in 44% of test cases. It is clear that even before further improvement by the genetic algorithm, including the probabilistically chosen waypoint sequences into the genetic algorithm for initial population creation makes a better path than a deterministic greedy algorithm most of the time.

Finally, Figure 2d shows the total number of planner calls for each algorithm as a function of the number of waypoints. Both the genetic and greedy algorithms demonstrate polynomial time complexity (they grow more slowly than Cn^2 for some C , as shown) and the brute force solution grows at least exponentially. The genetic algorithm also requires significantly more computation than the greedy. These results support expected trends.

Conclusions

This work has demonstrated the usage of approximation algorithms to generate feasible waypoint sequences for an energy-constrained rover in the lunar poles, where time-varying shadows greatly affect the science mission environment. While guaranteeing optimal solutions in such a scenario would require an algorithm with greater-than-polynomial worst-case complexity, the approximation algorithms tested in this research run in polynomial time. For ten waypoints or fewer, both the deterministic greedy and the genetic approximation algorithms achieved 80% or better of the optimal route value in all cases tested, and the genetic algorithm achieved 100% of the optimal value in all but one case. For 20 to 100 waypoints, this work did not compute

the optimal value, but demonstrated that both deterministic greedy and genetic algorithms could find effective solutions. Here, the genetic algorithm improved the route value over the deterministic greedy algorithm in a majority of cases.

On closer examination, improvements in route value over the deterministic greedy solution were found for two-thirds of the test cases just by executing the initialization process for the genetic algorithm, which runs a probabilistic version of the greedy algorithm. From there, the genetic algorithm improved over its initial conditions in 44% of the test cases. It is possible that the deterministic or probabilistic greedy algorithms found the optimal solution in some cases, making it impossible to improve further. This would be especially prevalent for smaller numbers of waypoints. However, for larger cases, it is more likely that this lack of improvement indicates that the genetic algorithm, as implemented here, may find itself caught in local minima based on the initialization phase. Several options for each of the steps in a genetic algorithm have been presented in prior work; this research examined a few of them. Future work could find a set of existing (or new) genetic algorithm options that would better explore the search space and be less likely to become fixed in a local minima. Particularly, different methods of crossover and mating may help diversify the gene pool and find more optimal solutions.

Other future work could include modification of the heuristic function used in the greedy algorithm to more accurately model a specific mission's parameters. Also, while time windows were not handled explicitly in the cases tested in this work, the software was developed with the ability to handle explicit time windows. This could be useful if, for example, a mission objective involved taking images under certain illumination conditions, but more testing would be needed to determine how effectively the algorithms presented here handle explicit time windows.

The algorithms explained in this paper offer a strong baseline for further studies regarding efficient multi-goal path-planning on planetary surfaces with time-varying shadows. Genetic algorithm approaches offer particular promise. Since a mission's destination will be determined far in advance, an initial mission plan can be computed by a genetic algorithm with parameters similar to those tested here. Re-planning during a mission will have to occur quickly, but due to the generational nature of a genetic algorithm, it can be stopped at any point after initialization with a feasible solution. Thus, a genetic algorithm approach offers a good balance between optimization and computation time.

Acknowledgments

Acknowledgement of funding sources will be added after review.

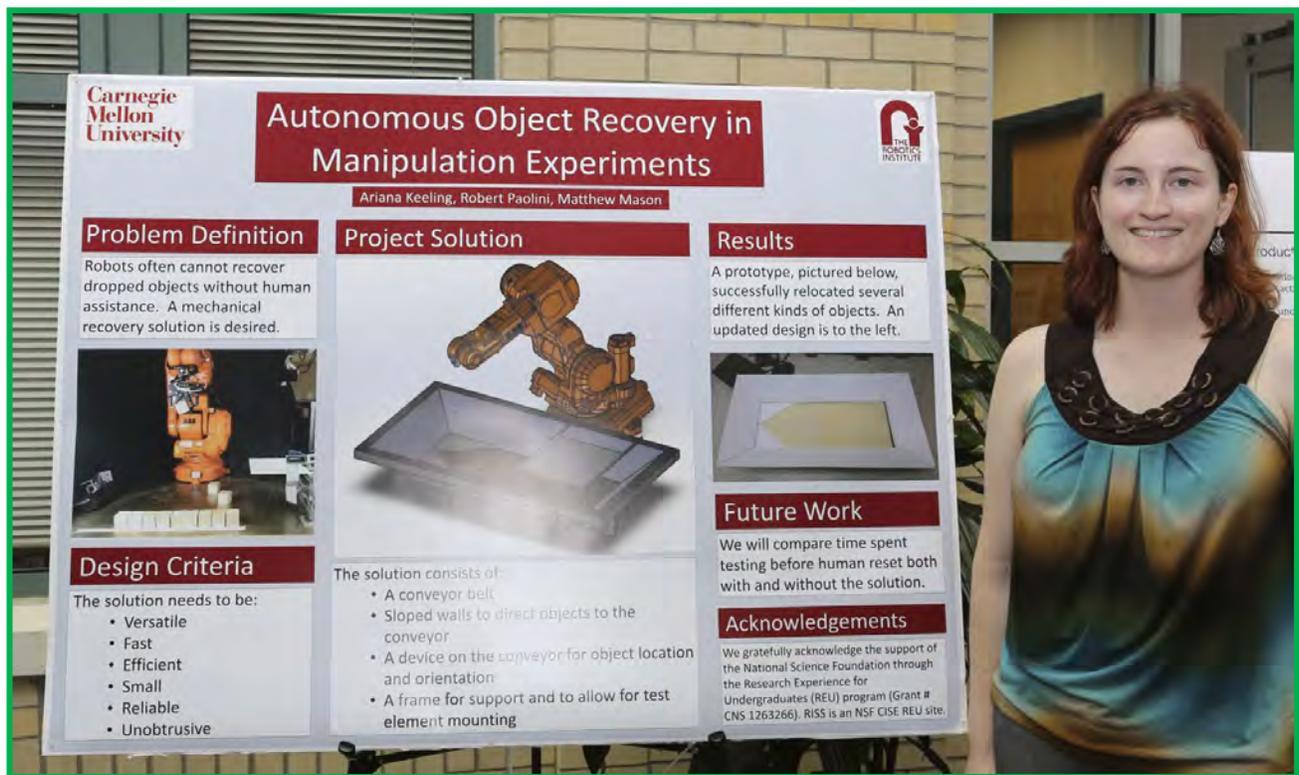
References

Barrett, A.; Bass, D.; Laubach, S.; and Mishkin, A. A. 2009. Retrospective snapshot of the planning processes in mer operations after 5 years. In *International Workshop on Planning and Scheduling for Space*.

- Cunningham, C.; Jones, H. L.; Kay, J.; Peterson, K. M.; and Whittaker, W. L. R. 2014. Time-dependent planning for resource prospecting. In *International Symposium on Artificial Intelligence, Robotics and Automation in Space*.
- Duque, D.; Lozano, L.; and Medaglia, A. L. 2015. Solving the orienteering problem with time windows via the pulse framework. *Computer & Operations Research* 54:168–176.
- Fomin, F. V., and Lingas, A. 2002. Approximation algorithms for time-dependent orienteering. *Information Processing Letters* 83(2):57 – 62.
- Garcia, A.; Arbelaitz, O.; Vansteenwegen, P.; Souffriau, W.; and Linaza, M. T. 2010. Hybrid approach for the public transportation time dependent orienteering problem with time windows. In *Hybrid Artificial Intelligence Systems*. Springer. 151–158.
- George, J.; Mattes, G.; Rogers, K.; Magruder, D.; Paz, A.; Vaccaro, H.; Baird, R.; Sanders, G.; Smith, J.; Quinn, J.; et al. 2012. Resolve mission architecture for lunar resource prospecting and utilization. In *Lunar and Planetary Science Conference*, volume 43, 2583.
- Gunawan, A.; Lau, H. C.; and Yuan, Z. 2014. *A Mathematical Model and Metaheuristics for Time Dependent Orienteering Problem*. Helmut-Schmidt-Univ., Univ. der Bundeswehr Hamburg.
- Kantor, M. G., and Rosenwein, M. B. 1992. The orienteering problem with time windows. *The Journal of the Operational Research Society* 43(6):629–635.
- Karbowska-Chilinska, J., and Zabielski, P. 2014. Genetic algorithm solving the orienteering problem with time windows. In *Advances in Systems Science*. Springer. 609–619.
- Keller, C. P. 1989. Algorithms to solve the orienteering problem: A comparison. *European Journal of Operational Research* 41(2):224–231.
- Mennell, W. K. 2009. *Heuristics for Solving Three Routing Problems : Close-Enough Traveling Salesman Problem, Close-Enough Vehicle Routing Problem, Sequence-Dependent Team Orienteering Problem*. PhD thesis, University of Maryland.
- Mishkin, A.; Limonadi, D.; Laubach, S.; and Bass, D. 2006. Working the martian night shift - the mer surface operations process. *Robotics Automation Magazine, IEEE* 13(2):46–53.
- Otten, N. D.; Jones, H. L.; Wettergreen, D. S.; and Whittaker, W. L. 2015. Planning routes of continuous illumination and traversable slope using connected component analysis. In *International Conference on Robotics and Automation*.
- Saha, M.; Sanchez-Ante, G.; Roughgarden, T.; and Latombe, J.-C. 2006. Planning tours of robotic arms among partitioned goals. *International Journal of Robotics Research* 25(3):207–223.
- Smith, D. E. 2004. Choosing Objectives in Over-Subscription Planning. In *Intl. Conf. on Automated Planning and Scheduling*.
- Tompkins, P. 2005. *Mission-Directed Path Planning for Planetary Rover Exploration*. Ph.D. Dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.
- Vansteenwegen, P.; Souffriau, W.; and Van Oudheusden, D. 2011. The orienteering problem: A survey. *European Journal of Operational Research* 209(1):1–10.
- Wang, X.; Golden, B. L.; and Wasil, E. A. 2008. Using a genetic algorithm to solve the generalized orienteering problem. In *The vehicle routing problem: latest advances and new challenges*. Springer. 263–274.
- Zabielski, P.; Karbowska-Chilinska, J.; Koszelew, J.; and Ostrowski, K. 2015. A genetic algorithm with grouping selection and searching operators for the orienteering problem. In *Intelligent Information and Database Systems*. Springer. 31–40.

Ariana Keeling

RISS 2015



Carnegie Mellon University

Autonomous Object Recovery in Manipulation Experiments

Ariana Keeling, Robert Paolini, Matthew Mason

Problem Definition
Robots often cannot recover dropped objects without human assistance. A mechanical recovery solution is desired.



Project Solution



Results
A prototype, pictured below, successfully relocated several different kinds of objects. An updated design is to the left.



Design Criteria
The solution needs to be:

- Versatile
- Fast
- Efficient
- Small
- Reliable
- Unobtrusive

Future Work
We will compare time spent testing before human reset both with and without the solution.

Acknowledgements
We gratefully acknowledge the support of the National Science Foundation through the Research Experience for Undergraduates (REU) program (Grant # CNS 1263266). RISS is an NSF CISE REU site.

Autonomous Object Recovery in Manipulation Experiments

Ariana Keeling, Robert Paolini, and Matthew T. Mason

Abstract—Robotic manipulators often drop the objects they are working with, requiring human intervention and recovery. The problem requires a mechanical solution that will help the robot to recover a dropped object without any human intervention. The mechanism needs to be quick, reliable, and able to accommodate multiple kinds of objects and experiments. The chosen solution is designed to be mounted under a platform the robot can use for experiments. It has sloped aluminum walls to direct a falling object to a conveyor belt. The robot can recover the object from the conveyor end. This should allow a robot to perform autonomous experiments for extended periods of time with no human input.

I. INTRODUCTION

Robotic manipulators sometimes need to attempt risky maneuvers to determine what will work and what will not. A failed attempt is useful as a data point; however, often a robot cannot recover a dropped object, requiring human intervention to reset the experimental workspace before the robot can continue. This human interaction with the robot undermines the autonomy of the learning system, and can make machine learning experiments, which require thousands of data points, difficult to finish.

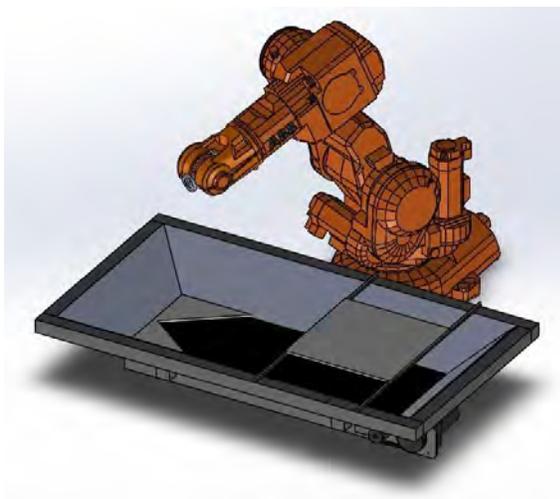


Figure 1: SolidWorks model of complete solution

This paper presents a mechanical solution to the object reset problem (Figure 1). The solution consists of a conveyor belt surrounded by four sloped aluminum walls. When an object falls, it is directed by the walls onto the conveyor where it travels to an area easily accessible by the robot. This solution has few moving parts, allowing for simplicity and taking advantage of the mobility of the robot. In addition, the small size of the solution increases its versatility.

Research supported by NSF Grant # CNS 1263266, for REUs.

Ariana Keeling was with the Robotics Institute Summer Scholars program at Carnegie Mellon University (ariana.keeling@gmail.com).

Robert Paolini and Matthew T. Mason are with the Robotics Institute at Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213 (rpaolini@cmu.edu, matt.mason@cs.cmu.edu)

II. RELATED WORK

There are several manipulation-based labs that use various methods to reset failed experiments. In the Manipulation Lab at Carnegie Mellon University, an aluminum ramp has been used to redirect falling highlighter pens into a hopper; in addition, for experiments in which the robot manipulates (and frequently drops) a block, eleven extra blocks are lined up in a known location for the robot to use once it drops its previous block (Figure 2). The aluminum ramp worked well but does not provide a recovery option, and the multiple-block solution only allows for around two hours of continuous experimentation. At Yale, objects used in manipulation experiments are attached to strings; these strings are looped through a hole in the manipulator and attached to weights which pull dropped objects back into the manipulator [6]. This is a robust solution; however, it does not allow for the desired level of mobility.

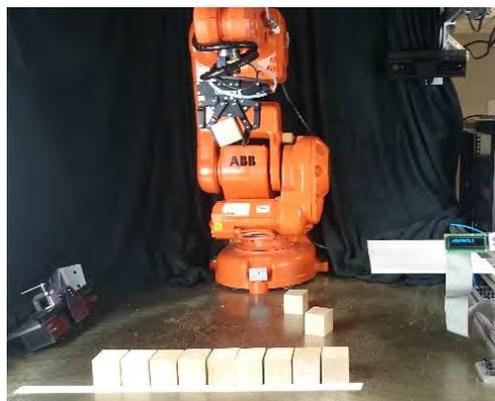


Figure 2: Multiple block solution

Another source of inspiration for this solution was Hoops the Basketball Robot, on display at the Carnegie Science Center of Pittsburgh [5]. Hoops is an industrial arm which is programmed to throw basketballs through a hoop. It then must recover the ball autonomously in order to continue. Hoops uses a preprogrammed motion to sweep the floor around it, pushing the ball toward the back of its workspace. Once the ball falls into a divot in the floor, Hoops can pick it back up and throw it again. The design space presented in this paper is similar to that of Hoops: both are required to recover a single object in a robust and efficient manner. However, Hoops will only ever recover a single kind of object, and has a large workspace around it, whereas our solution space has little area around the robot and must be able to handle a variety of objects. In addition, all of the motions of the Hoops robot are preprogrammed, while the robot used for this project is capable of sensing the location and orientation of the object.

The problem of automated object reset is similar to the problem of relocating and orienting parts in industry. Automated assemblies use parts feeders to move objects

from an initial location, often a hopper, to an assembly location. There are many kinds of parts feeders, most of which are optimized for the location and orientation of a single kind of object in a particular assembly environment. The most common parts feeders are vibratory parts feeders, which use vibratory motion to propel parts along the path from the hopper to the robot, and centrifugal parts feeders, which use centripetal force to propel objects up a ramp [2]. While both of these methods are widely used, parts feeders of this sort are often designed specifically to locate and orient a single kind of object, and can take a long time to optimize. In addition, these parts feeders excel at moving many objects at once; however, for the given application only a single part is being relocated at any one time. Even when these feeders are designed to be versatile, they can only accommodate "families" of parts, and still require a fair amount of design and optimization to get even that low level of versatility [1].

More recently, manufacturers have prioritized parts feeders that can handle a variety of objects with minimal redesign. In particular, agile manufacturing processes and flexible parts feeders have been the subject of a lot of recent industrial automation research. These versatile parts feeders often use conveyor belts as transport options for parts because they can accommodate a wide variety of differing sizes and features. These feeders also use vision systems to identify the location and orientation of objects for a mobile robot to grasp. However, many do not attempt to orient or locate objects on a conveyor prior to grasping, likely due to the large number of objects being moved at a time. Also, many involve multiple-conveyor systems to lift the parts, carry them to the robot, and return parts that are in an undesirable location or orientation to the start point. Our system requires the part to succeed on its first go around, and the initial location for the object is broad and encompasses the whole conveyor system. [3]

The use of magnets, as in [4], was considered as a method to catch, hold, or relocate objects; this idea was ultimately rejected due to a desire to handle objects made of many different materials without modification. However, the design is notable in that it is robust, can be relatively small, and is capable of handling a variety of objects that are below a certain weight and have ferrous elements.

III. SOLUTION

We designed solutions to the above problem using an IRB-140 mounted on a large table; however, many of the solutions below are applicable or adaptable to various workspaces.

A. Design Criteria

The problem presented several solution requirements. The first and most important requirement is *versatility*: accommodating multiple kinds of objects in different experiments. The solution must also be *small* and *inconspicuous* to maximize robot workspace. In addition, in order to maximize time spent experimenting, the device was desired to be *fast*, with the full cycle of the solution occurring within twenty seconds of failure recognition. The solution

must also be *reliable*. This put some priority on solutions with few moving parts, little power input, and low sensing requirements.

B. Potential Solutions

Six alternative solutions developed over the course of the project are detailed below and compared against the problem criteria.

Vacuum



Figure 3: Vacuum solution

The vacuum solution consists of a hose vacuum suspended above the robot workspace and held out of the way by a spring or other elastic structure. When the robot misplaces its object, it uses the vacuum to pull the object back to a recoverable location. This can involve a uniform, whole-area sweeping solution or a camera-driven pinpoint operation. This solution is particularly useful for issues with an object in close proximity to another object or a wall; however, in the case of general object recovery, the solution may not be as efficient or reliable as desired. In addition, the vacuum nozzle would need significant modification to be able to handle various kinds of objects. This solution would work well in a cluttered workspace where proximity more than obstruction is an obstacle to object retrieval, and where objects are large and flat.

Movable Walls



Figure 4: Translational walls solution

A movable walls solution would involve two sets of opposite walls moving in tandem along tracks to push a lost object back to a designated location on the robot workspace. This solution is efficient and reliable; however, its versatility is hampered by the linear motion of the walls, which would restrict the ability to place stationary testing elements in the center of the workspace. This could be alleviated by designing the experiments to have suspended elements instead of mounted ones. In addition, it could be difficult to design the walls to handle small objects, which could be easily wedged underneath the moving structures. Nonetheless, this solution is significant in that it requires no

sensing at all to locate the object, and is very useful in terms of clear workspaces and larger objects.

Magnets

This solution involves an elevated platform with a line of magnets ringing the edge. When the robot fails, the object will adhere to the nearby magnets, preventing the object from falling from the robot workspace. This solution was outstanding in terms of its small size and efficiency; however, its versatility and reliability were questionable at best, and all objects would need to be modified in some way such that the magnets could catch them if they fell. This solution would work well with a robot that was working exclusively with ferrous materials.

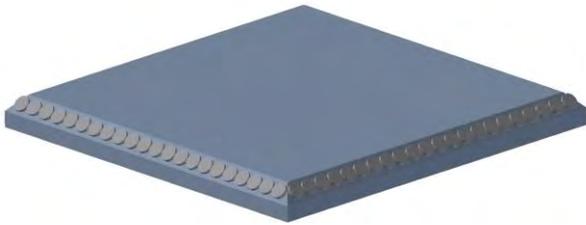


Figure 5: Magnets solution

Air Jets

A solution involving air jets would function in a similar manner to an air hockey table. In this solution, an air tube wraps around the robot workspace with valves at calculated intervals on the inside wall. Implementation can happen in two ways: either a vision system can determine which valves need to open in order to move the object, or the valves can be opened in a set pattern to ensure object relocation if minimal sensing is desired. This solution is unobtrusive, and could work very well for smaller objects, but may be unreliable and inefficient. An air jets solution would be best implemented on smaller setups, with limited workspaces and light objects, and would be most efficient when implemented with a vision system.



Figure 6: Air jets solution

Cloth Walls

This solution involves an elevated platform with loose cloth walls attached on all sides. When the robot fails and the object falls off the platform, the object is caught by the cloth walls and can be retrieved by lifting the top of the walls, tautening the cloth and forcing the object back onto the platform. This solution would be efficient and versatile; however, it would take a lot of careful design to ensure that

an object would not get tangled in the walls, and there would be an issue with walls around the area of the robot, since the robot is an arm. Cloth walls would work well for a robot which is mounted directly above the workspace, and would be most easily implemented for a round workspace.



Figure 7: Cloth walls solution

Collection Area

A collection area solution involves a raised platform mounted over a bin with sloped walls. The bottom of the bin is a small area where the object location is relatively known. The object can be retrieved from the collection area in several different ways: for an area directly under the platform, a cart can be used to bring the object back up to the platform, while with an offset area the robot can fetch the object directly. The appeal of this solution is that it has few moving parts and can accommodate a wide variety of different objects. However, because the object must slide down the walls, there is a minimum height requirement for gravity to overcome the coefficient of friction. This solution would work best in experimental setups where the height of the platform is not an issue, or where the solution can be mounted underneath the workspace.

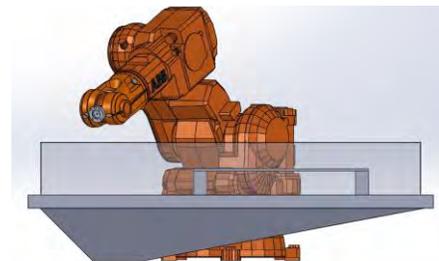


Figure 8: Offset collection area solution

Potential Solution Analysis

An evaluation of the potential solutions against the design criteria is shown in Table 1. A plus indicates that the solution fulfills the requirement, while a minus indicates that the solution does not fill the requirement. A zero indicates that the solution fulfills the requirement, but does not do it well. The pluses and minuses are summed for a final value. When compared with the various design criteria, the two solutions that stood out were the movable walls and collection area solutions. The collection area solution was chosen due to its lack of moving parts. Several designs were considered, with an offset collection area ultimately prototyped and tested. Due to the height requirement of the solution, the design was later scrapped in favor of a combination collection area and conveyor belt solution.

Table 1: Evaluation of potential solutions against design criteria

	Fast	Reliable	Versatile	Small
Vacuum	-	+	+	+
Moveable Walls	+	0	+	+
Magnets	+	-	-	+
Air Jets	0	-	0	+
Cloth Walls	+	-	+	-
Collection Area	+	+	+	+

C. Chosen Solution

The final solution design (Figure 1) uses sloped aluminum walls and a conveyor belt to relocate wayward objects. The conveyor sits low to the ground and is mounted in front of the robot. The conveyor has 80/20 framing, which is used to mount the sloped aluminum walls. The walls form a trapezoidal shape to catch objects on a variety of trajectories and direct them to the conveyor. Additional 80/20 framing wraps around the top of the walls both to mount and support the walls and to allow for mounting of test elements above the solution. This provides a reconfigurable workspace that will suit many different kinds of experiments. In addition, taller vertical walls can be mounted along the top outside of the solution in the case of experiments involving greater amounts of force.

At the end of the conveyor belt, there is an inverse wedge mounted to assist with location and partial orientation of an object. This wedge can be moved or replaced to accommodate different objects or to allow for a greater level of orientation depending on the needs of the experimenter and the particular scenario. In addition, the back wall of the solution is detachable to allow for further versatility. The ability to locate and orient objects allows for a simple vision sensor to give the robot enough data to collect a fallen object.

When the robot is experimenting and drops its object, it first determines whether the object is somewhere retrievable. If it is not visible, the robot signals the conveyor to turn on. The conveyor remains on for a length of time equivalent to the amount of time the belt needs to cross half the conveyor. At the end of this length of time, the conveyor belt turns off, and a camera mounted above the orientor takes a picture of the end of the conveyor. The robot uses this image and the known solution configuration to find the exact location and orientation of the object, and can then retrieve the object to continue experimentation.

IV. RESULTS

A prototype was used to evaluate the solution. In order to test the ability of the solution to relocate objects, a 50 mm wooden cube was dropped off of an experimental setup 50 times (Figure 9). Of these 50 trials, 4 trials were failures, with the block falling outside the workspace. In addition, 3 trials resulted in the block falling on the orienting wedge (Figure 10). In certain cases, the robot may be able to retrieve objects from on top of the wedge; however, it is a problem which will need to be addressed. These results are quantified in Table 2.

Table 2: Results of trials for 50 mm wooden cube

Trials	Successes	Failures	On Wedge
50	43	4	3

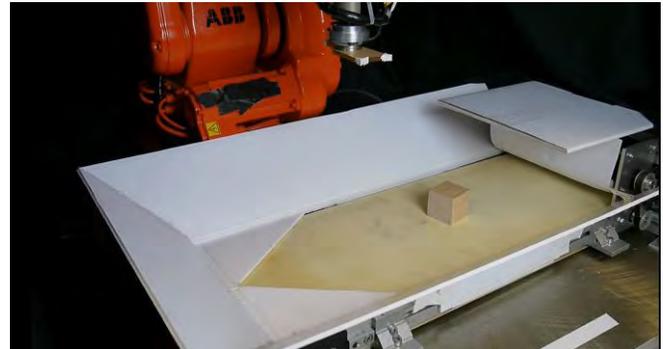


Figure 9: Prototype solution successfully relocating a block

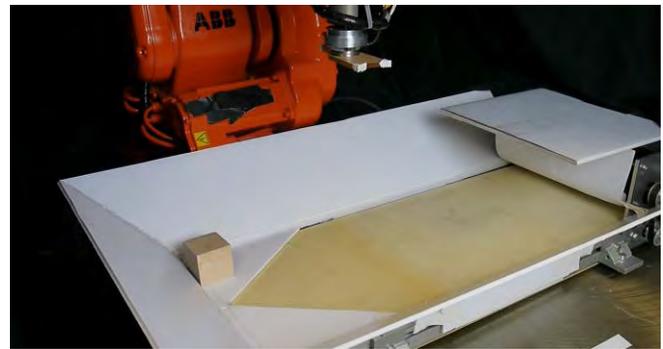


Figure 10: Trial in which the block landed on the orienting wedge

To test the versatility of the solution, a variety of objects (Figure 11) were dropped off of the same experimental setup on average five times each. The results were similar to those of the wooden block, with most trials resulting in a successful relocation of the object. A few objects, such as the largest triangle and the allen wrench, were large enough to get stuck in between the platform and the wall of the solution. The ball ended up on the orienting wedge four out of five trials. For all other objects, the majority of the trials resulted in successes. A summation of these results is presented in Table 3.



Figure 11: A variety of objects used to test the versatility of the solution

Table 3: Results of trials for various objects

	Trials	Successes	Failures	On Wedge
Small Cube	4	3	1	0
Tall Square	5	5	0	0
Rectangle	4	4	0	0
Big Triangle	5	4	1	0
Small Triangle	4	4	0	0
Triangle with Hole	6	6	0	0
Cylinder	6	6	0	0
Bridge	4	4	0	0
Nail Polish Remover	6	6	0	0
Die	5	4	0	1
Clamp	5	5	0	0
Allen Wrench	6	5	1	0
Marker	6	6	0	0
Tape	5	3	0	2
Ball	6	1	0	5

The results of the tests with the prototype revealed a number of issues that will need to be resolved for the final implementation. The issue of the size of objects can be resolved by designing experimental workspaces to allow for the required amount of clearance between the workspace and the aluminum walls. A larger version of the solution could be made to accommodate still larger objects. In addition, the likelihood of an object falling outside of the solution is small, and can be easily prevented by raising the height of the walls.

The largest and most pressing issue is the tendency of objects to land on the orienting wedge. While these objects may be recoverable in some cases, this recoverability will depend on the end effector on the robot and the size and position of the object. One possible solution to this issue is to allow the robot to grasp a flat piece which it can use to sweep the object off of the orientor and back onto the conveyor. Another is to redesign the orienting wedge to accommodate an object: for example, a taller wedge would prevent a ball from rolling down the conveyor and onto the orientor. In any case, a large amount of force is required to propel an object onto the wedge or outside the solution, so many experiments can be run while this problem is under consideration.

V. CONCLUSION

This paper presents a robust mechanical solution to the object reset problem. This solution will allow for extended

periods of autonomous manipulation experiments and machine learning.

Some further work is required to finalize elements of the design. The orienting wedge at the end of the conveyor must be optimized for a variety of objects. Implementation of backwards motion in the conveyor could allow for better orientation of objects, as explored in [7]. In addition, fences mounted above or along the conveyor, as in [8] and [9], could be used to help orient or locate objects.

ACKNOWLEDGMENT

Special thanks to the Robotics Institute Summer Scholars program at Carnegie Mellon University, which provided the opportunity for this research.

REFERENCES

- [1] G. P. Maul and J. L. Goodrich, "A Methodology for Developing Programmable Part Feeders," *IIE Transactions*, vol. 15, issue 4, pp. 330-336, 1983.
- [2] P. Khanna *et al.*, "Design and Fabrication of Centrifugal Hopper Feeder and Study of Factors Affecting It," *IJAEA*, vol. 2, issue 4, pp. 45-49, 2009.
- [3] G. C. Causey and R. D. Quinn, "Design of a Flexible Parts Feeding System," *Proc. IEEE Robot. Automat.*, vol. 2, pp. 1235-1240, Apr. 1997.
- [4] G. Singh *et al.*, "Magnetic Elevating Hopper Feeder: Design, Fabrication and Analysis of Various Parameters Affecting it," *IJAEA*, vol. 2, issue 4, pp. 56-61, 2009.
- [5] Carnegie Science Center, 'Hoops', 2015. [Online]. Available: <http://www.carnegiesciencecenter.org/exhibits/roboworld-meet-robots-hoops/>. [Accessed: 15- Jul- 2015].
- [6] Personal communication with Aaron Dollar of the GRASP Lab at Yale
- [7] S. Akella and M. T. Mason, "Parts Orienting with Shape Uncertainty," *Proc. IEEE Robot. Automat.*, vol. 1, pp. 565-572, May 1998.
- [8] J. Wiegley *et al.*, "A Complete Algorithm for Designing Passive Fences to Orient Parts," *Assembly Automation*, vol 17, issue 2, pp. 129-136, 1997.
- [9] K. M. Lynch, "Inexpensive Conveyor-Based Parts Feeding," *Assembly Automation*, vol. 19, issue 3, pp. 209-215, April 1999

Multi Modal Pose Fusion for Monocular Flight

Arbaaz Khan
Autonomy Researcher

Motivation

Autonomous flight through cluttered environments



Platform



- Optical flow is used for pose estimation.
- Pose from optical flow is noisy and fails over bad textured terrain.
- Semi-Dense Visual Odometry is used for perception and it also provides pose. However, it fails when unable to track points between keyframes.
- Need to fuse data from the two sources to get accurate and reliable pose.

Approaches

Lie Algebraic Averaging

- N images parametrized by $N-1$ independent motions. There exist $(N(N-1)/2)$ image pairs each of which provide a constraint on global motion.
- Lie algebra representation provides efficient system for averaging this overdetermined system of equations.
- Disadvantage – both SDVO and flow tracker are equally weighted in the average.

Kalman Filter

- Inputs weighted by their variance and then averaged.
- We find results are only marginally better.

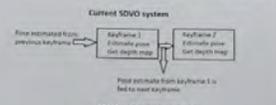
RMSE over 45 meters



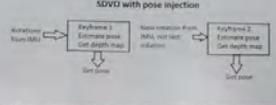
IMU pose injection for SDVO

- SDVO tracks points to get depth maps. It tracks points between multiple frames.
- Instead of using rotations estimated from camera frame, we inject rotations from IMU which are more accurate.

Current SDVO system



SDVO with pose injection



Average number of resets (instances where SDVO crashes) during flight time reduced by 42%.

Performance with Pose injection

Pose Evaluation

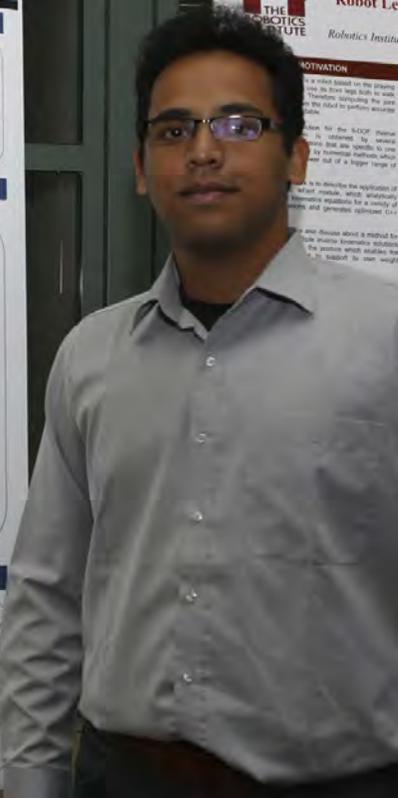


Number of resets over 1000 m



Acknowledgement

I would like to thank the following people who have helped me on this project: Debadeepta Dey, Sar Zeng, Shreyansh Datta, Rachel Burcin, and the entire RISS team for their wonderful support through the summer.



65

Multi Modal Pose Fusion For Monocular Flight

Arbaaz Khan
School of Electronics and
Communications Engineering
Manipal Institute of Technology
Manipal, India
Email: arbaazkhan2@gmail.com

Shreyansh Daftry
Robotics Institute,
Carnegie Mellon University,
Pittsburgh, USA
Email: daftry@cmu.edu

Martial Hebert
Robotics Institute
, Carnegie Mellon University,
Pittsburgh, USA
Email: hebert@ri.cmu.edu

Abstract—State estimation using a monocular camera is a very popular topic of research in robotics. Some of the common methods involve fusing pose from two sources using a kalman filter or a lie algebraic representation for solving the global motion constraints. Some systems even use visual inertial systems to estimate robust pose by taking into consideration the pose from an inertial measurement system (IMU). This work extends the idea of a tightly coupled visual inertial system to a photometric error based method of semi dense visual odometry. With extensive experimentation we prove our visual inertial system for semi dense visual odometry is better than using visual odometry alone. We also show robustness of our method and compare our performance to the existing state of the art tightly coupled visual inertial systems that exist in an outdoor environment.

Keywords—Semi Dense Visual Odometry, Monocular Flight, .

I. INTRODUCTION

Pose estimation is a diverse problem in robotics that has been addressed several times. The problem of estimating three dimensional pose has become relatively more important due to the recent advances in simultaneous localization and mapping leading to higher degree of autonomy in robots. The challenge with pose estimation arises due to its vastly different settings in different scenarios. Estimating pose of an object for a robotic arm manipulator is significantly different from estimating pose of an unmanned aerial vehicle. Within this paper we develop a system to efficiently estimate pose of an unmanned aerial vehicle using a visual inertial using only a monocular camera and an on board inertial measurement unit (IMU).

II. IMU COUPLED SEMI DENSE DEPTH MAP ESTIMATION

In this section we describe our approach for state estimation and describe the problems that arise from our approach. We take this further and apply some of the standard practices for robust state estimation and show how these do not improve the net results by a significant amount. Using this as motivation, we develop a system for robust pose estimation using the on board inertial measurement unit while at the same time provide accurate semi dense depth maps.

A. Pose Estimation

In our current system we use a downward facing camera for state estimation using the famous Lucas-Kanade optical flow (flow tracker) algorithm [1]. The front facing camera is primarily used for depth propagation. However, since we use semi dense visual odometry, it can also be used for estimating

pose. Using optical flow on a downward facing camera tends to be tricky. Previous studies on the performance of the optical flow algorithm for different textures and scenarios have shown the performance to vary hugely under varied scenarios [2]. If the texture of the ground is good, results obtained are generally very close to ground truth. But if the texture on the ground is bad as is the case for a lot of forest environments, state estimates from optical flow are noisy. For sake of brevity we shall use the term pose estimate and state estimate interchangeably. To resolve the problems arising from noisy optical flow data, we propose using the pose from the front facing camera too. We try two of the most standard approaches to fuse the poses from both the camera's. The first approach uses lie algebraic averaging to produce globally consistent motion as is described in the work of Govindu *et. al* [3]. To summarize this approach, consider the 4x4 euclidean motion matrix $\mathbf{M} \in SE3$ defined as

$$\mathbf{M} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \quad (1)$$

where $\mathbf{R} \in SO3$ and $\mathbf{t} \in \mathbb{R}^3$. This \mathbf{M} is converted to lie space by using the log operator. The result is another 4x4 matrix

$$\mathbf{m} = \begin{bmatrix} \Omega & u \\ 0 & 0 \end{bmatrix} \quad (2)$$

where $\mathbf{m} \in \mathfrak{se}(3)$. Algebraic averaging is then performed on this lie algebraic representation of motion. Now consider two motions matrices \mathbf{M}_1 and \mathbf{M}_2 with \mathbf{M}_1 being estimated from the projections of the downward facing camera and \mathbf{M}_2 being estimated from the projections of front facing camera. \mathbf{M}_2 is rotationally aligned and scaled to the same world as \mathbf{M}_1 . These are then converted to lie space and then averaged out according to the algorithm described in [3]. However, we observe that the results are only marginally better (see Fig 2). These results are attributed to the fact that semi dense visual odometry does not work well if there is a huge rotation between two consecutive frames. More specifically this problem arises due to the fact that if there is a large enough rotation between two consecutive frames, the points that are being tracked from for which the inverse depth maps are generated are lost. This problem can be specifically seen in Fig. 1. It must be noted that orientation of SDVO must be realigned to that of the flow tracker every time tracking is lost to minimize drift.

To offset the reset problem, a standard variance weighted input averaging technique (a Kalman Filter based fusion) was also tried [4]. Treating the flow tracker as a known prior distribution $\mathcal{N}(d_p, \sigma_i^2)$ and the SDVO as a noisy observation

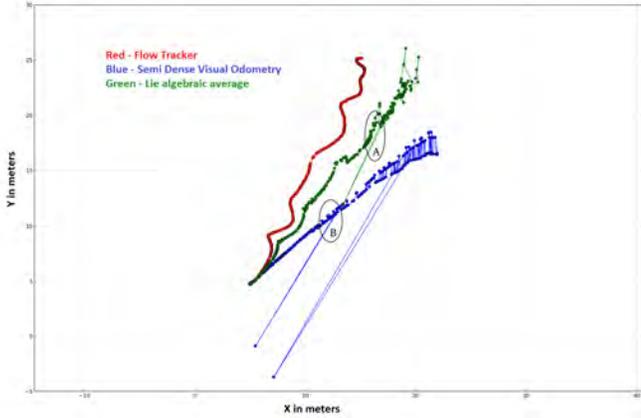


Fig. 1. Loss of tracking in SDVO causes huge jumps in the pose which is also reflected in the lie algebraically averaged pose. This can be clearly seen at instances A and B

$\mathcal{N}(d_q, \sigma_j^2)$, the posterior is given by

$$\mathcal{N}\left(\frac{\sigma_i^2 d_q + \sigma_j^2 d_p}{\sigma_i^2 + \sigma_j^2}, \frac{\sigma_i^2 \sigma_j^2}{\sigma_i^2 + \sigma_j^2}\right) \quad (3)$$

However, even with Kalman Filtering our results tend to be only marginally better. Evaluation of pose estimates was done by bench marking with a real time kinematic (RTK) GPS [5] with an accuracy up to 5 cm.



Fig. 2. Pose benchmarking using a RTK GPS over a distance of 45 meters.

It is also noticed that at points where tracking is lost for the entire frame, there is a loss in depth map propagation. At such instances the entire SDVO pipeline needs to be reset and this is highly undesirable for fast monocular flight through a cluttered environment. Using this as motivation, in the next section we describe a mechanism where we couple our IMU with the SDVO to ensure robust pose and depth estimates.

B. IMU coupling for semi dense depth map estimation

We build on the work of Engel *et. al* [6]. To summarize their approach, given a semi-dense inverse depth map for the current image, the camera pose of the new frames is estimated using direct image alignment: given the current map $\{I_M, D_M, V_M\}$, the relative pose $\xi \in SE(3)$ of a new frame I is obtained by directly minimizing the photometric error

$$E(\xi) := \sum_{x \in \Omega_{D_M}} \|I_M(x) - I(w(x, D_m(x), \xi))\|_\delta \quad (4)$$

where $w : \Omega_{D_M} \times \mathbf{R} \times SE(3) \rightarrow \omega$ projects a point from the reference frame image into the new frame. D and V denote mean and variance of the inverse depth, and $\|\cdot\|_\delta$ is the Huber norm to account for outliers. The minimum is computed using iteratively re-weighted Levenberg-Marquardt minimization [7]. The warp function $w(x, T)$ as described in [8] is defined as

$$w(x, T) = \pi(T\pi^{-1}(x, Z(x))). \quad (5)$$

where $\pi^{-1}(x, Z)$ represents the inverse projection function and T represents the transformation matrix, for a rigid body motion g

$$\mathbf{T} = \begin{bmatrix} \mathbf{R}_{3,3} & \mathbf{t}_{3,1} \\ 0 & 1 \end{bmatrix} \quad (6)$$

Since T is over parametrized representation of g and has twelve parameters while g has only six degrees of freedom, we use ξ , the twist coordinate representation for T where

$$(\xi) = \log_{SE(3)}(T) \quad (7)$$

The photometric error is minimized iteratively and the pose update step can be represented as

$$(\xi)^{n+1} = \delta(\xi)^n \circ (\xi)^n \quad (8)$$

where $\delta(\xi)^n$ is calculated by solving for the minimum of a Gauss Newton second order approximation of E . However, as seen from our previous experimentation, the projection function fails to map a point from one image I_1 to another I_2 if the rotation is very large. An alternate method for solving this problem can be to consider a very high frame rate of the camera. But this has its own limitations in terms of power and computational complexity. Thus, we propose a IMU injection to overcome this limitation. The traditional approach for coupling an IMU with a visual system is to add a term for the IMU in the energy function [9]. While this works well when the error in state measurement is minimized, it is relatively more complex to incorporate an IMU term within the SDVO's photometric error minimization. Our approach involves putting the injection term inside the optimizer instead. The update step has been modified to not accept rotations calculated from the previous keyframe. Instead the rotations from the IMU are fed into the optimizer for depth propagation. Therefore to summarize, the rotations that are needed to represent rigid body motion in Eqn. (6) are directly taken from the IMU instead of taking the pose estimate from the previous iteration of the optimizer. Also to ensure that the error is not propagated, we change the way optimization is done. This is better illustrated in Fig. 3

With the IMU injection into the optimizer, the pose drift is considerably lesser. In addition to this, the instances where SDVO would crash are reduced to by 55% over a distance of 1 km. This is seen in Fig. 4 The inverse depth map is propagated from frame to frame, once the pose of the frame has been determined and refined with new stereo depth measurements. Based on the inverse depth estimate d_0 for the pixel, the corresponding 3D point is calculated and projected into the new frame and assigned to the closest integer pixel position providing the new inverse depth estimate d_1 . In the original work it is assumed that the camera rotation is small, and the new inverse depth map can be approximated by

$$d_1(d_0) = (d_0^{-1} - t_z)^{-1} \quad (9)$$

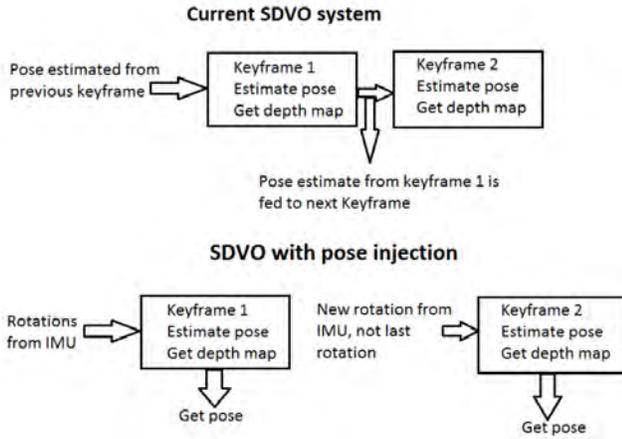


Fig. 3. A comparison of the original SDVO pipeline and our version modified for monocular flight.

where t_z is the camera translation along the optical axis. Now, for each frame, after the depth map has been updated, a regularization step is performed by assigning each inverse depth value the average of the surrounding inverse depths, weighted by their respective inverse variance.

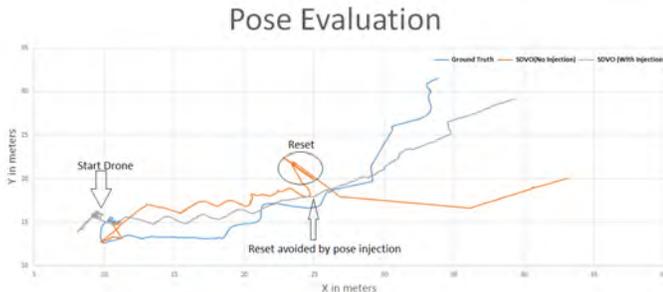


Fig. 4. Trajectory comparison using SDVO with and without injection. The blue line represents ground truth, orange line represents SDVO without pose injection and the grey line represents SDVO with pose injection.

III. DISCUSSION AND FUTURE WORK

It is evident from the results that our proposed method of injecting IMU rotations into the optimizer results in fewer instances where tracking is lost thereby leading to fewer instances where the depth map crashed and the whole pipeline must be restarted. However, even with our pose injection there are instances where the system still crashes. Most of these instances are when the autonomous mode is just switched on or off. Further, the pipeline fails to provide reasonable depth when the environment is not cluttered, such as small clearings in the woods. One might agree that in such situations it is redundant to track depth. However, the depth tracked is only in the field of view of the monocular camera and it is possible that obstacles could be present in the environment just outside the field of view of the vehicle and could contribute to a crash. In our future work we wish to address this problem by utilizing stored information from previous depth maps.

REFERENCES

- [1] B. D. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision." in *IJCAI*, vol. 81, 1981, pp. 674–679.
- [2] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *International journal of computer vision*, vol. 12, no. 1, pp. 43–77, 1994.
- [3] V. M. Govindu, "Lie-algebraic averaging for globally consistent motion estimation," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1. IEEE, 2004, pp. I–684.
- [4] O. Wijk, H. Christensen *et al.*, "Triangulation-based fusion of sonar data with application in robot pose tracking," *Robotics and Automation, IEEE Transactions on*, vol. 16, no. 6, pp. 740–752, 2000.
- [5] S. Edwards, P. Cross, J. Barnes, and D. Betaille, "A methodology for benchmarking real time kinematic gps," *Survey Review*, vol. 35, no. 273, pp. 163–174, 1999.
- [6] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *European Conference on Computer Vision (ECCV)*, September 2014.
- [7] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *Computer Vision–ECCV 2014*. Springer, 2014, pp. 834–849.
- [8] C. Kerl, J. Sturm, and D. Cremers, "Dense visual slam for rgb-d cameras," in *Proc. of the Int. Conf. on Intelligent Robot Systems (IROS)*, 2013.
- [9] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Vision-based state estimation for autonomous rotorcraft mavs in complex environments," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013, pp. 1758–1764.

Ratnesh Madaan

RISS 2015



Predicting orientations under manipulative actions

Ratnesh Madaan, Erol Sahin, Robert Paolini, Matthew T. Mason



Introduction

- What is the final orientation of an object when a robotic gripper does an action on it?
- How can we represent uncertainty over the space of rotations - $SO(3)$?

Consider a cube dropped by a gripper
 Predict the face on which it falls - Classification
 Predict the angular displacement - Regression

The Bingham Distribution

- Antipodally symmetric probability distribution over a hypersphere.
- Derived from a zero mean Gaussian on \mathbb{R}^n constrained to lie on S^n or \mathbb{P}^n .

$$f(\mathbf{x}; \Lambda, V) = \frac{1}{Z(\Lambda, V)} \exp\left(-\frac{1}{2} \mathbf{x}^T \Lambda \mathbf{x}\right)$$

$Z(\Lambda, V) = \int_{S^2} \exp\left(-\frac{1}{2} \mathbf{x}^T \Lambda \mathbf{x}\right) d\mathbf{x}$

$\Lambda = \text{diag}(\lambda_1, \lambda_2, \lambda_3)$

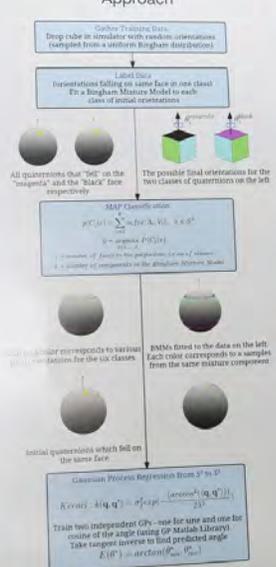
$V = \text{diag}(v_1, v_2, v_3)$

$\lambda_1 > \lambda_2 > \lambda_3$

$v_1 > v_2 > v_3$

For visualizing a distribution over quaternions, we can rotate a reference point (yellow) by samples from \mathcal{Q} .

Approach



Reduced problem for parallel grippers

Parallel jaw gripper
 Possible orientations are in a "plane" \mathcal{Q}^p

Reduce problem
 To n circular regression problems $\mathcal{Q}^1 \rightarrow \mathcal{Q}^n$
 % of faces of polyhedron

Train per action, per gripper
 GP doesn't perform well. The NPCM library for regression (Quaternions, Matrices and local linear regression)

Simulation Results

- Dropping with random orientation
 Classification accuracy - 96.53%
- Parallel jaw gripper (NPCM)

Classifier	Accuracy
1 nearest neighbor	96.53%
Mean	33.25
Std Dev	21.75
Max	132.94

Classifier	Accuracy
1 nearest neighbor	96.53%
Mean	2.240e-02
Std Dev	0.0033
Max	0.0111

Predicting orientations under manipulation actions

Ratnesh Madaan¹, Robert Paolini¹, Erol Sahin², Matthew T. Mason¹

Abstract—We present a method for predicting the resulting orientation of an object when a manipulative action is performed on it. We represent orientations using unit quaternions and model uncertainty via Bingham distributions on the unit quaternion hypersphere. Assuming that the object is stabilized at the beginning and the end of manipulation either because it is gripped or because it rests on a surface, we use prior interaction data to regress the final orientation from its initial orientation. We validate the approach by predicting the outcome of a cube dice throw in simulation. The problem of $SO(3)$ regression is broken down into two parts - first classification is performed via a maximum likelihood method to predict the face on which the dice will fall on, which is followed by a Gaussian Process regression from the initial quaternion(Q) to find the predicted final angle(S^1). Further, the simplified case of a two finger, rigid, parallel jaw gripper is discussed, where the initial orientation can be reduced to an angle(S^1), and the Nadaraya-Watson estimator is used for non-parametric circular regression in order to predict the final orientation.

I. INTRODUCTION

Manipulation can be considered as changing the pose of the object in the inertial coordinate frame. Modeling the effects of a manipulative action using kinematic and dynamic models in an analytic way usually fails to fully capture the nonlinear interactions between the object, the manipulator, and the environment. Such physics based simulations require precise definition of the three entities, their possible interactions, the constraints involved, and repeating everything for a different scenario. We argue that modeling a robotic manipulation action can be best achieved by learning a map from the initial pose of an object to its final pose using data collected from actual interactions and present a statistical inference based approach.

The full 6D pose of a rigid object is represented with a position and an orientation, and lies in $SE(3)$ which is defined as $R^3 \times SO(3)$, where $SO(3)$ is the special Orthogonal group in which rotations reside. Learning a mapping from the initial pose of an object to its final pose requires the use of a parametric or nonparametric regression method $SE(3) \mapsto SE(3)$. However such a method has not been developed yet, thanks to the peculiarities of $SO(3)$.

In this study, we propose a regression method for predicting the orientation of an object given its initial orientation, without solving the full $SO(3) \mapsto SO(3)$ problem using assumptions that typically hold in robotic manipulation. Specifically, we assume that the object is at rest and its orientation is constrained by its shape as well the plane on

which the object rests or which is defined by contact patches of the gripper holding it.

For instance, consider the final orientations of a cube dice dropped on a table with a random initial orientation and for the sake of simplicity, from a constant height. The possible resulting orientations of the dice forms 6 clusters, each corresponding to the cube lying on one of its face. For each of these clusters, the orientation of the cube can be specified with only one rotational degree of freedom. The cluster can be seen as a submanifold of $SO(3)$ and we use Bingham distributions over quaternions to define the same. Similarly, there will be 6 clusters of initial orientations as well, each defined by the criteria that the corresponding final orientations belong to the same cluster or ‘lie’ on the same face. In this sense, a mapping from the initial rotation of the dice to its final orientation can be split into two problems: a classification problem that maps the initial rotation to one of the resulting orientations’ clusters (outcome of the dice throw) and a mapping from the initial orientation of the dice to the final 1-D orientation of the dice, defined as $SO(3) \mapsto F^f \times S^1$ where F^f is the finite set of different clusters in the final orientation space, and S^1 is the 1-D circular variable.

Moreover, if the dice was being held by a two-fingered, rigid, parallel jaw robotic gripper, its initial orientations would be subject to the same characteristics as well¹. In such a case, the problem is reduced to $F^i \times S^1 \mapsto F^f \times S^1$, where F^i is the finite set of different clusters in the initial orientation space.

Generally, we propose to fit a Bingham Mixture Model to each class of initial orientations of the object which share a trait in their final orientations. The shared trait of a class of initial orientations could be as simple as a common face in the final orientation in the case of a dice or something more complex for a different action and object shape. We use a maximum likelihood approach to classify the test sample, i.e. predicting the face on which the dice will land on if it is dropped with the orientation corresponding to the test quaternion, which is followed by Gaussian Process regression ($Q \mapsto S^1$) with a kernel that utilizes the quaternion arc distance to evaluate the similarity metric. For the case of the two-fingered jaw gripper, we use circular regression ($S^1 \mapsto S^1$) by utilizing the NPCirc library, which implements nonparametric kernel density estimation via the adaptation of the Nadaraya-Watson and local linear estimators.

¹Robotics Institute at Carnegie Mellon University, Pittsburgh, PA, USA

²Kovan Research Lab, Dept. of Computer Eng., Middle East Technical University, Ankara, Turkey

¹We neglect cases such as jamming where the faces of the dice are not aligned by the inner surfaces of the gripper.

This paper is organized as follows. Section II presents a quick review of quaternions, the Bingham distribution and mixture models (BMM), and various ways to visualize them. Section III explains our classification-cum-regression approach to predict the final orientation of a cube when it is dropped in a simulator. Section IV tackles the case of two-fingered gripper, and circular regression. Section V discusses how the data was gathered, and presents results. Section VI explores related work. Finally, section VI presents the conclusion, discussion, and future work.

II. QUATERNIONS, BINGHAM DISTRIBUTION, AND VISUALIZATIONS

There are various ways to represent $SO(3)$ including Euler angles, rotation matrices, axis-angle representation and unit quaternions, and all of them have their pros and cons. However, we choose quaternions as they are: (a) not as ambiguous as Euler Angles apart from antipodal symmetry, (b) do not suffer from the Gimbal lock problem, (c) easier to compose as compared to the axis-angle representation, (d) allow for relatively simple interpolation using SLERP, and (e) allow for faster computation. For our case, an even stronger motivation for using quaternions is that the Bingham distribution captures the antipodal topology of the quaternion space perfectly. To top that, the Bingham is a member of the exponential family and is the maximum entropy distribution on the hypersphere [1]. There has been a recent surge of interest in and the revival of Bingham distributions due to the Glover's work [2] and they provide a promising model for rotational uncertainty.

The Bingham distribution is derived from a multivariate Gaussian constrained to lie on a hypersphere. It is antipodally symmetric, and taking into account the facts that the mapping from quaternion to the rotation space is two to one and that antipodal quaternions $(q, -q)$ represent the same 3-D rotation/orientation, it is intuitive to use it to represent uncertainty over the rotation space along with unit quaternions. One could also consider a hemisphere of S^3 and use another distribution like Von Mises-Fisher, for instance, to model the rotation space; however it is convenient to consider the whole hypersphere coupled with a Bingham and not bother with the discontinuities at the equator. Here, it is worth mentioning that the VMF distribution is also a maximum entropy distribution over a hypersphere, and it could be a viable avenue for modeling rotations [1]. Mathematically, the Bingham distribution is represented as:

$$f(\mathbf{x}; \Lambda, V) = \frac{1}{F(\Lambda)} \exp\left\{\sum_{i=1}^d \lambda_i (\mathbf{v}_i^T \mathbf{x})^2\right\} \quad (1)$$

$$= \frac{1}{F(\Lambda)} \exp\{\mathbf{x}^T V \Lambda V^T \mathbf{x}\} \quad (2)$$

where \mathbf{x} is a unit vector on the hypersphere S^d , Λ is a $d \times d$ diagonal matrix of concentration parameters $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_d (\leq 0)$, V is the matrix of the eigenvectors (v_1, v_2, \dots, v_d) of the distribution, and F is the normalization constant. As explained in [2], the $(d+1)^{th}$ eigenvalue λ_{d+1} (and

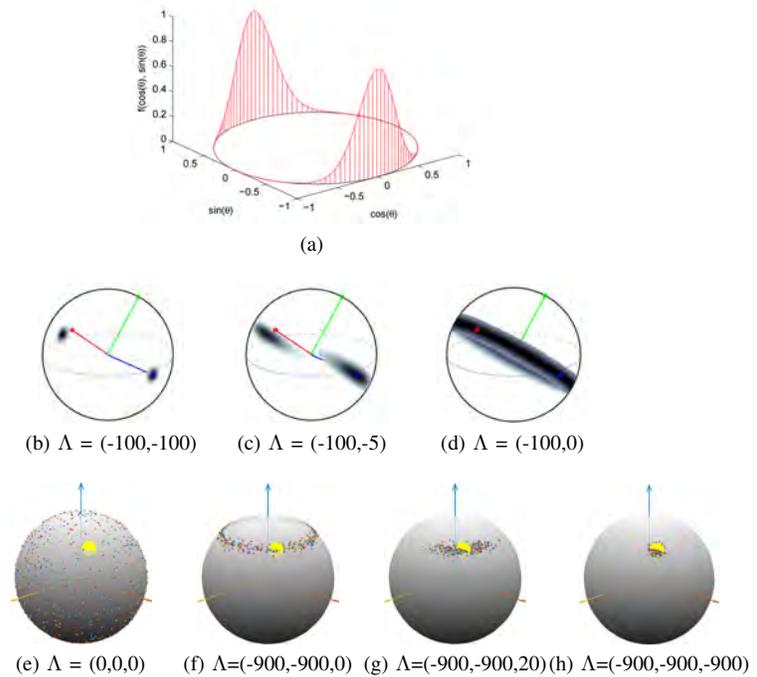


Fig. 1: Example Bingham distributions on S^1 [14], S^2 [2] and S^3 , with varying concentration parameters.

its corresponding eigenvector v_{d+1}) are omitted by adding $-\lambda_{d+1}$ to all eigenvalues, without affecting the distribution.

Fig. 1 shows some example Bingham distributions : (a) over a circle, (b)-(d) over a sphere, with varying concentration parameters; (e)-(h) are visualizations of the quaternion Bingham distribution (referred to as qBingham, and whose eigenvectors will be referred to as eigenquaternions in the remaining text) inspired by the heat plots of [7] as explained later. Intuitively visualizing a qBingham is not that easy as quaternions reside on a 4D-hypersphere. We now explore various ways to visualize a qBingham (or quaternions themselves) by taking samples from it and plotting them. When visualizing a 4D object in 3D, there is always an inherent loss of 1 degree of freedom of information. Hence, we need to add information to the 3D visualization as explained in the upcoming paragraphs.

Consider a cube (Fig. 2(a) and (b)), whose faces are colored for easy reference. Imagine the possible orientations of the cube when it's lying on one of its faces, say with the black face on top, and then with the pink face on top.

Fig. 2(c) and (d) shows the axis-angle representation, with the line segment being parallel to the axis of rotation and the length being proportional to the angle of rotation about the axis, with the cases of the black and pink faces on top respectively. Here we are adding information by encoding the length of the segment to be proportional to the magnitude of the angle of rotation about it. For negative rotations, we reverse the direction of axis. Generally, this representation is not intuitive as one has to imagine a complex rotational transform from the identity orientation (Fig. 2(a)) in just one step, as compared to three transformations in the Euler

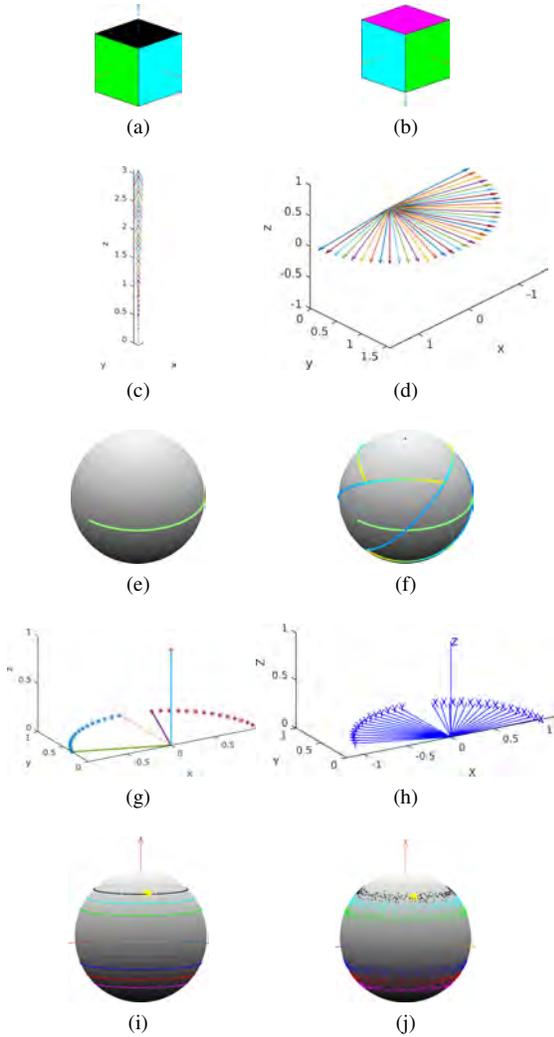


Fig. 2: Visualizing quaternions and qBinghams.

angle representation. However, Fig. 2(c) and (d) are easier to understand; in the former the axis is the same but the angle is changing, in the latter it's vice-versa. If we think of the analogous transformations when the cube is lying on one of the other four face, things become complex as both the axis and angle will change.

Fig. 2(e) and (f) are similar to Fig. 2(c) and (d), but here information is added via color coded points on a sphere - red represents the maximum angle, and blue the minimum. Fig. 2(e) plots the same data as (d) where all the samples have the same angle, but different axis. The analogue for Fig. 2(c) will be just a couple of dots at the poles of the sphere and hence we omit it. Fig. 2(f) is the representation of all possible orientations when the cube is lying on one of its faces. These 6 clusters are submanifolds in the $SO(3)$ space. For the remaining four faces, the axis-angle duo form a pattern as shown. The angle of rotations changes gradually as the axis changes. These plots were made by modifying a function of [3], which also visualizes the Hopf fibrations of the 4D-sphere. In [4], Glover refers to a similar visualization (without

the sphere) of orientated local features as Q-Images.

One could also simply view the final coordinate axes corresponding to a quaternion relative to the reference frame via an animation. Fig. 2(g) shows a snapshot of the same. It plots the endpoints of the coordinate axes for each quaternion while showing the axis corresponding to the current quaternion in the animation iteratively. This is obtained using the Matlab quaternion class [5], which we also use for dealing with quaternions to obtain the results presented in the later sections. Fig. 2(h) is the same visualization using the Robotics Toolbox [6], but this time instead of their endpoints, the axes themselves are retained after each animation.

Fig. 2(i)-(j), and Fig. 1(e)-(h) are inspired from Reidels heat plot visualization of the qBingham [7]. A reference point on a sphere, shown by the big yellow marker, is repeatedly rotated about its center by many sampled quaternions (here 1000 in number), and the final positions of the reference point, which are bound to lie on the sphere itself, are visualized as the dotted markers. The eigenquaternions of the Bingham in Fig. 1(e)-(h) are $(0,1,0,0)$; $(0,0,1,0)$; $(0,0,0,1)$ and their respective concentration parameters can be seen in the captions. -900 is a computational limit in [3], which corresponds to zero uncertainty about a particular eigenquaternion. The final position dots are randomly colored, and the color doesn't carry any meaning in Fig. 1. Fig. 2(i) shows various orientations of the cube when it's lying on a face and plots the same data as in Fig. 2(f). Fig. 2(j) shows samples from a Bingham Mixture Model fitted to the quaternions of (i) and (f), and is discussed in the coming paragraphs. It should be noted that Fig. 2(i) or (j) are not equivalent to the cases where qBingham of Fig. 1 would have $\Lambda = (0, -900, -900)$ or $(-900, 0, -900)$ and so on, which represent uncertainty about X and Y axis respectively. The catch is that the cube is lying on a face and not freely rotating in air. Hence, we have horizontal rings of Fig. 2(i), instead of analogous vertical rings.

An implicit caveat of this visualization is that if the axis of rotation passes through the reference point itself, then those rotations can't be visualized as the reference point will stay at its initial position no matter what the angle is. To circumvent this issue, we use two or more reference points, each lying on a different sphere. This visualization, is the most intuitive and quickest way to interpret a qBingham or quaternions themselves. It also helps to look at the animations corresponding to Fig. 2(g) and (h) along with this visualization.

As the complexity of a set of quaternions increases, we need a more flexible distribution. Akin to Gaussian Mixture Models, we use finite Bingham Mixture Models which are a weighted sum of component qBinghams mathematically represented as [4]

$$f(x; \mathbf{B}, \boldsymbol{\alpha}) = \sum_{i=1}^k \alpha_i f(x; B_i) \quad (3)$$

To fit a BMM, one needs to find the number of clusters, the weight of each component and the parameters of the

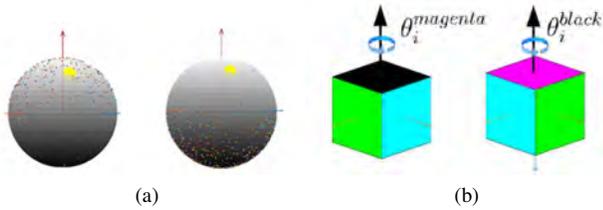


Fig. 3: Labeling the data. (a) Left and right show the initial orientations of the cube with which it fell on the magenta and the black faces respectively, and hence belong to the same class (b) The final orientations of the cube when it is dropped with initial orientations plotted in (a).

Bingham components. We use the Bingham Statistics Library for this (and all other Bingham distribution related purposes) [3], which implements the BMM-SAC algorithm, a fast, greedy, sample consensus based approach, to fit a BMM to a given dataset [4]. Another modified-EM based approach is explained in [8]. To visualize a BMM, we use the Fig. 1 qBingham visualization, but now color the samples to indicate samples from the same Bingham component, as shown in Fig. 2(j).

III. APPROACH

Consider a cube being dropped with a random initial orientation. We sample 6000 quaternions from a uniform Bingham distribution (Fig. 1(e)), drop the cube in a simulator with a random sample, note the final orientation of the cube and label all the initial quaternions corresponding to which the cube fell on a single face as belonging to the same class (Fig. 3). Labeling is done by fitting a BMM to the final orientations. One can identify each face by the eigenquaternions of the qBingham components. Then we fit a BMM to each class of initial quaternions. This helps us to define a sub-manifold of $SO(3)$ for which the cube is likely to fall on the same face. For classification, we take a maximum likelihood approach, assigning classes to the test sample based on which set it is most likely to belong to, or in our case, which face it is most likely to fall on.

The next step is regression, or predicting the cubes final orientation, given which face it is predicted to be lying on. We need to regress from the quaternion corresponding to the initial orientation q_i to the quaternion corresponding to the final orientation q_f . But as all the final quaternions are actually various orientations of the cube when its lying on one of its faces, instead of regressing from Q to Q , we can regress from Q to S^1 . The next section describes in detail how we extract the response (training target) angle from Bingham distribution (by using one of its eigenquaternion). We use Gaussian Processes regression to achieve the same and utilize the quaternion arc kernel, which uses the arc length distance between quaternions to calculate the distance metric [15].

$$k(q, q') = \sigma_f^2 \exp\left(-\frac{\text{arccos}^2(\langle q, q' \rangle)}{2\lambda^2}\right) \quad (4)$$

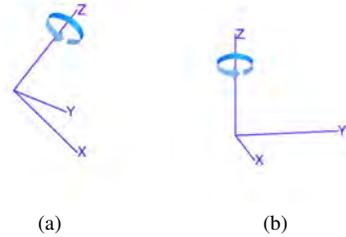


Fig. 4: (a) Q_{tilt} : set of orientations in a tilted ‘plane’, and (b) Q_z , which is Q_{tilt} after multiplying each element with q_{trans}

The training output is an angle $\theta \in R$, which has a discontinuity at $\theta = 2\pi$, or in other words 1° is quite close to 359° while they are far apart on the number line. To avoid this problem, we train two GPs - one with the training targets being the sine of the angle, and the other being trained the cosine of the angle, or what [9] refers to as the SinCos-GP. Hence, two sets of independent hyperparameters are learned. Once we have the predicted values of the sine (θ_{sin}^*) and cosines (θ_{cos}^*), we use the four-quadrant tangent inverse to evaluate the predicted angle.

$$E(\theta) = \arctan(\theta_{sin}^*, \theta_{cos}^*) \quad (5)$$

IV. CIRCULAR REGRESSION FOR PARALLEL JAW GRIPPER

Consider a two-finger, rigid, parallel jaw gripper holding a cube. In this case, all the possible orientations of the cube are in a ‘plane’ relative to the gripper (Fig. 4(a)). Let’s call this set Q_{tilt} . This is akin to the case of cube lying on the floor, and the orientation can be reduced to an angle from a full quaternion here as well. To reduce these orientations to a single angle θ , we can change the frame of reference such that the normal vector to the ‘plane’ of orientations is now the positive Z axis, and hence the corresponding quaternions (denoted by the set Q_z) are of the form $[\cos(\theta/2), 0, 0, \sin(\theta/2)]^T$ as shown in Fig. 4(b). Once this is done we can easily find the ‘predictor’ angle(training input) and similarly, the ‘response’ angle(training target) by taking cos or sin inverse of the first or the fourth component.

It is a one to one mapping, and the same rotational transformation (q_{trans}) is needed to be multiplied for each element from Q_{tilt} to get the corresponding element in Q_z . If we fit a BMM to Q_{tilt} , it will have just a single component due to the ‘plane’ of rotation, whose concentration parameters will be $(-900, -900, 0)$. The eigenquaternion corresponding to the third, zero-valued parameter of the component Bingham is then the inverse of the required q_{trans} . Hence, we multiply all $q_{tilt} \in Q_{tilt}$ by $q_{trans} = \{q_{eigen, \lambda=0}\}^{-1}$ to get the corresponding $q_z \in Q_z$, thereby changing the frame of reference.

$$q_{tilt} * \{q_{eigen, \lambda=0}\}^{-1} = q_z, \forall q_{tilt} \in Q_{tilt} \text{ and } q_z \in Q_z \quad (6)$$

Finally, we use the NPCirc library, which implements circular kernel density estimation, using the local linear smoothing estimator for non-parametric circular regression in order to predict the final orientation [10].

V. SIMULATION EXPERIMENTS AND RESULTS

A. Random initial orientation

To verify the model, we drop a cube with random initial orientations (6000 samples from a uniform qBingham) from a constant height in the V-REP simulator and note the final orientations. The confusion matrix is shown the Table I and we can predict with 90.53% accuracy which face the cube will fall on. As evident from the Table I, the algorithm does not predict the cube to land on the opposite face in all cases. (The trace is 5990 as 10 samples could not be labeled properly due to simulation or labeling error.)

For the correctly classified orientations, we proceed with regression, and the errors are showed in Table II. We compare GP regression with two other methods as shown. 1-nearest neighbour is a naive way to do regression, where the predicted output is the final orientation when the cube is dropped with the orientation corresponding to the closest quaternion in the training sample to the test sample (using distance metric $d = 1 - |q_1 \cdot q_2|$, where ‘.’ refers to the quaternion dot product). SLERP (spherical linear interpolation) is similar, but tries to interpolate the resultant orientation of the two closest quaternions. The last column shows the result by the suggested Gaussian Process regression method. It is clear to see that GP outperforms the other two by a large margin.

	Face 1	Face 2	Face 3	Face 4	Face 5	Face 6	Total
Face 1	936	0	17	25	14	41	1033
Face 2	0	884	40	54	37	15	1030
Face 3	18	8	900	0	16	14	956
Face 4	25	15	0	896	25	18	979
Face 5	43	18	19	19	933	0	1032
Face 6	10	28	27	21	0	874	960
Total	1032	953	1003	1015	1025	962	5990

TABLE I: Confusion matrix for the case of random initial orientations

B. Parallel jaw gripper

Table III shows the results by dropping a cube from random initial orientations, which are constrained to lie in a ‘plane’, hence simulating the scenario when a cube is dropped by a two finger, parallel jaw gripper. We analyze

	1-nearest neighbour	SLERP	Gaussian Process
Mean	13.29	15.79	0.61
Std Deviation	17.51	41.72	2.69
Maximum	132.99	179.98	7.22

TABLE II: Errors (in degrees) with random initial orientation

(Angle with horizontal)	Face(0°)	Edge(45°)	Edge(30°)
Classification	100%	56.72%	81.09%
Mean	2.2459e-04	1.21	0.29
Std Deviation	0.0033	2.87	1.39
Maximum	0.0111	10.85	5.09

TABLE III: Errors (in degrees) for the case of parallel jaw gripper

this case for three different situations - the first case is where the normal to the ‘plane’ of orientations is the positive Z axis and hence the cube is generally landing on a face, the second case is where the cube is always landing on an edge but the angle with the horizontal is 45°, and the third case is where the angle is 30°. For the initial orientations where one of the faces is parallel to the floor, we get 100% accuracy in prediction as edge collisions are minimal. The accuracy is close to 50% if the angle of the ‘plane’ of rotations with the horizontal is 45° and the cube always falls on its edge. In this case, collisions are aplenty and accuracy is low. Finally, if we decrease the angle with the horizontal to 30°, the classification accuracy increases. In all three cases, GP regression performs fairly well as can be seen from the mean, standard deviation and maximum error, which are evaluated using a Matlab circular statistics toolbox [11].

VI. RELATED WORK

The closest work to ours in terms of the ideology and the framework presented is [12], which presents a statistical framework using two distributions (to combine local and global information) in order to predict the motion of a body after it comes in contact with another body, and is able to demonstrate generalization to a new shape for which the model wasn’t trained. The philosophy of [12] resonates with ours in that simulators can not generalize to new scenarios without explicitly adding information about the new object and the environment. [13] uses Mixture of Projected Gaussians, along with dual quaternions to model 6D pose uncertainty, while tackling the pose estimation problem. [2] explains the quaternion Bingham distribution and its properties in detail, along with filters for 3D-orientations, and validates the theory on a ping pong playing bot based on an object detection system presented in the thesis. The results in the thesis provide evidence of Bingham distributions being a suitable choice for depicting rotational uncertainty.

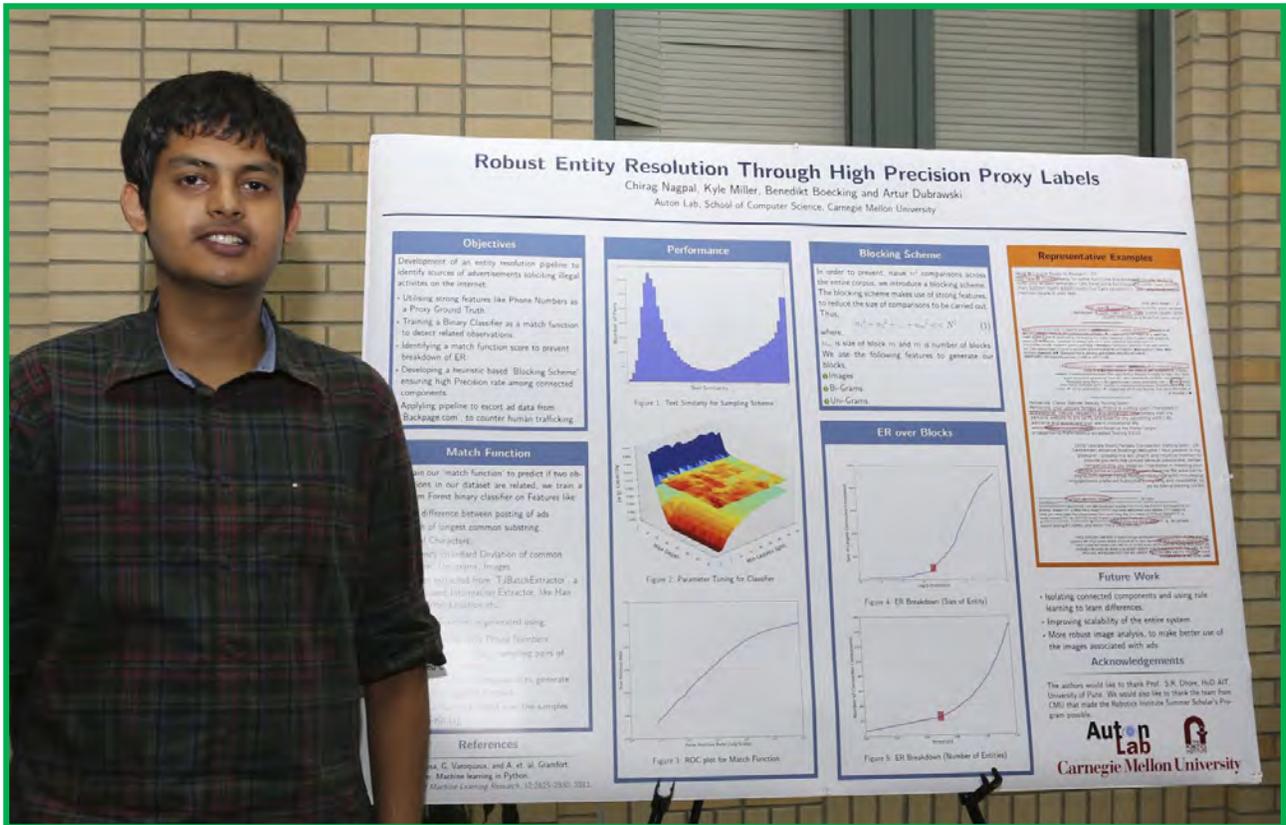
VII. CONCLUSION

A method for predicting orientations under manipulative actions is described. It is proposed that a physical action on an object can be modeled probabilistically using the Bingham distribution, given a lot of data - preferably with random initial orientations, or those encompassing the relevant domain of testing data for the particular action - in order to incorporate all possible resulting orientations of the object. The case of a two finger parallel jaw gripper is analyzed as well. Experiments are carried out in a simulator and the

regression errors are calculated using circular statistics. It is proposed that simple actions like dropping, pushing, and throwing an object could be seen as building blocks of a library of more complex grasping behaviour. Future work includes looking at ways to extend our approach to predict 6D poses. One avenue could be using GMMs for predicting positions while retaining BMMs for orientations, and another points towards the unexplored territory of using Bingham distributions along with dual quaternions.

REFERENCES

- [1] Mardia, Kantilal Varichand. *Statistics of directional data*. Academic press, 2014.
- [2] Glover, Jared Marshall. *The quaternion Bingham Distribution, 3D object detection, and dynamic manipulation*. Diss. Massachusetts Institute of Technology, 2014.
- [3] J. Glover, *libbingham Bingham Statistics Library*, 2013. [Online]. Available: <http://code.google.com/p/bingham/>
- [4] Glover, Jared, Gary Bradski, and Radu Bogdan Rusu. "Monte Carlo Pose Estimation with Quaternion Kernels and the Bingham Distribution."
- [5] Mark Tincknell(2011). *quaternion.m*(<http://www.mathworks.com/matlabcentral/fileexchange/33341-quaternion-m>), MATLAB Central File Exchange. Retrieved June 28, 2015.
- [6] P.I. Corke, *Robotics, Vision & Control*, Springer 2011, ISBN 978-3-642-20143-1.
- [7] Riedel, Sebastian (2014) *Bayesian Orientation Estimation and Local Surface Informativeness for Active Object Pose Estimation*. Master's. DLR-Interner Bericht. 572-2014/29, 102 S.
- [8] Niezgodá, Stephen R., and Jared Glover. "Unsupervised learning for efficient texture estimation from limited discrete orientation data." *Metallurgical and Materials Transactions A* 44.11 (2013): 4891-4905.
- [9] Guerrero, Pablo, and J. Ruiz del Solar. "Circular Regression Based on Gaussian Processes." *Pattern Recognition (ICPR)*, 2014 22nd International Conference on. IEEE, 2014.
- [10] Oliveira, Mara, Rosa M. Crujeiras, and Alberto Rodriguez-Casal. "NPCirc: An R Package for Nonarametric Circular Methods." *JOURNAL OF STATISTICAL SOFTWARE* 61.9 (2014): 1-26.
- [11] Berens, Philipp. "CircStat: a MATLAB toolbox for circular statistics." *J Stat Softw* 31.10 (2009): 1-21.
- [12] Kopicki, Marek, Jeremy Wyatt, and Rustam Stolkin. "Prediction learning in robotic pushing manipulation." *Advanced Robotics*, 2009. ICAR 2009. International Conference on. IEEE, 2009.
- [13] Feiten, Wendelin, and Muriel Lang. "MPG-a framework for reasoning on 6 dof pose uncertainty." *ICRA 2011 Workshop on Manipulation Under Uncertainty*, Shanghai, China. 2011.
- [14] Gilitschenski, Igor, et al. "Unscented Orientation Estimation Based on the Bingham Distribution." (2013).
- [15] Lang, Michael, Oliver Dunkley, and Sandra Hirche. "Gaussian process kernels for rotations and 6d rigid body motions." *Robotics and Automation (ICRA)*, 2014 IEEE International Conference on. IEEE, 2014.



An Entity Resolution Approach to Isolate Instances of Human Trafficking Online

Chirag Nagpal*, Kyle Miller†, Benedikt Boecking†, and Artur Dubrawski†

*University of Pune, India

chiragnagpal_12102@aitpune.edu.in

†Auton Lab, School of Computer Science, Carnegie Mellon University

{mille856, boecking, awd}@andrew.cmu.edu

Abstract—Human trafficking is a challenging law enforcement problem, with a large amount of such activity taking place online. Given the large, heterogeneous and noisy structure of this data, the problem becomes even more challenging. In this paper we propose an entity resolution pipeline using a notion of proxy labels, in order to extract clusters from this data with prior history of human trafficking activity. We apply this pipeline to 5M records from *backpage.com* and report on the performance of this approach, challenges in terms of scalability, and some significant domain specific characteristics of our resolved entities.

Index Terms—human trafficking, machine learning, entity resolution.

I. INTRODUCTION

OVER the years human trafficking has grown to be a challenging law enforcement problem. The advent of the internet has brought the problem in the public domain making it an ever greater societal concern. Prior studies [5] have leveraged computational techniques to this data to detect spatio-temporal patterns, by utilising certain features of the ads. Certain studies [4] have utilised machine learning approaches to identify if ads could be possibly involved in human trafficking activity. Significant work has also been carried out in building large distributed systems, to store and process such data, and carry out entity resolution to establish ontological relationships between various entities. [7]

In this paper we explore the possibility of leveraging this information to identify sources of these advertisements, isolate such clusters and identify potential sources of human trafficking from this data using prior domain knowledge.

In case of ordinary Entity Resolution schemes, each record is considered to represent a single entity. A popular approach in such scenarios is a ‘merge and purge’ strategy whereas records are compared and matched, they are merged into a single more informative record, and the individual records are deleted from the dataset. [2]

While our problem can be considered a case of Entity Resolution, however, escort advertisements are a challenging, noisy and unstructured dataset. In case of escort advertisements, a single advertisement, may represent one or a group of entities. The advertisements hence might contain features belonging to more than one individual or group.

The advertisements are also associated with multiple features, including Text, Hyperlinks, Images, Timestamps, Locations etc. In order to featurize characteristics from text we use the regex based information extractor described in [4], based on the GATE framework [3]. (The reader is directed to [4] for

a complete description of the fields extracted and performance measures thereof.) This allows us to generate certain domain specific features from our dataset, including, the aliases, cost, location, phone numbers, specific URLs, etc of the entities advertised. We use these features, along with other generic text, the images, etc as features for our classifier. The high reuse of similar features makes it difficult to use exact match over a single feature in order to perform entity resolution.

We proceed to leverage machine learning approaches to learn a function that can predict if two advertisements are from the same source. The challenge with this is that we have no prior knowledge of the source of advertisements. We thus depend upon a strong feature, in this case Phone Numbers, which can be used as proxy evidence for the source of the advertisements and can help us generate labels for the Training and Test data for a classifier. We can therefore use such strong evidence as to learn another function, which can help us generate labels for our dataset, this semi-supervised approach is described as ‘surrogate learning’ in [8]. Pairwise comparisons result in an extremely high number of comparisons over the entire dataset. In order to reduce this, we use a blocking scheme using certain features.

The resulting clusters are isolated for human trafficking using prior expert knowledge and featurised. Rule learning is used to establish differences between these and other components. The entire pipeline is shown in Figure 1.

II. ENTITY RESOLUTION

A. Definition

We approach the problem of extracting connected components from our dataset using pairwise entity resolution. The similarity or connection between two nodes is treated as a learning problem, with training data for the problem generated by using ‘proxy’ labels from existing evidence of connectivity from strong features.

More formally the problem can be considered to be to sample all connected components $H_i(V, E)$ from a graph $G(V, E)$. Here, V , the set of vertices ($\{v_1, v_2, \dots, v_n\}$) can be considered to be the set of advertisements and E , $\{(v_i, v_j), (v_j, v_k), \dots, (v_k, v_l)\}$ is the set of edges between individual records.

We need to learn a function $M(v_i, v_j)$ such that $M(v_i, v_j) = Pr((v_i, v_j) \in E(H_i), \forall H_i \in H)$

The set of strong features present in a given record can be considered to be the function ‘ a ’. Thus, in our problem, $a(v)$ represents all the phone numbers associated with v .



Fig. 1: Entity Resolution Pipeline

The union of the set of strong features for all records is ‘A’. Thus $A = \bigcup a(v_j), \forall v_j \in V$. Here, $n(A) \ll n(V)$

Let us consider a graph $G^*(V, E)$ defined on the set of vertices V , such that $(v_i, v_j) \in E(G^*)$ if $n(a(v_i) \cap a(v_j)) > 0$

Let us consider the set, H^* of all the of connected components $\{H_1^*(V, E), H_2^*(V, E), \dots, H_n^*(V, E)\}$ defined on the graph $G^*(V, E)$

Now, function P is such that for any $p_i \in A$
 $P(p_i) = V(H_k^*), \text{ iff } p_i \in \bigcup a(v_j), \forall v_j \in (V(H_k^*))$

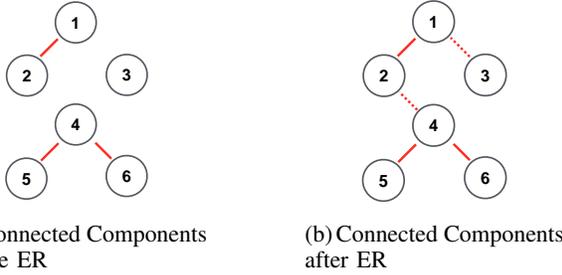


Fig. 2: On applying our match function, weak links are generated for classifier scores above a certain match threshold. The strong links between nodes are represented by Solid Lines. Dashed lines represent the weak links generated by our classifier.

B. Sampling Scheme

For our classifier we need to generate a set of training examples ‘T’,

T_{pos} is the set of positive training examples and T_{neg} is the set of negative training examples.

$$T_{pos} = \{F(v_i, v_j) \mid v_i \in P(p_i), v_j \in P(p_i), \forall p_i \in A\}$$

$$T_{neg} = \{F(v_i, v_j) \mid v_i \in P(p_i), v_j \notin P(p_i), \forall p_i \in A\}$$

In order to ensure that the sampling scheme does not end up sampling near duplicate pairs, we introduce a sampling bias such that for every feature vector $F(v_i, v_j) \in T_{pos}$, $a(v_i) \cap a(v_j) = \phi$

This reduces the likelihood of sampling near-duplicates as evidenced in Figure 3, which is a histogram of the jaccards similarity between the set of the unigrams of the text contained in the pair of ads.

$$JS(v_i, v_j) = \frac{n(\text{unigrams}(v_i) \cap \text{unigrams}(v_j))}{n(\text{unigrams}(v_i) \cup \text{unigrams}(v_j))}$$

We observe that although we do still end with some near duplicates ($JS > 0.9$), we have high number of non duplicates. ($0.1 < JS < 0.3$) which ensures robust training data for our classifier.

C. Training

To train our classifier we use a Random Forest classifier using Scikit.[6] Table I shows the most informative features

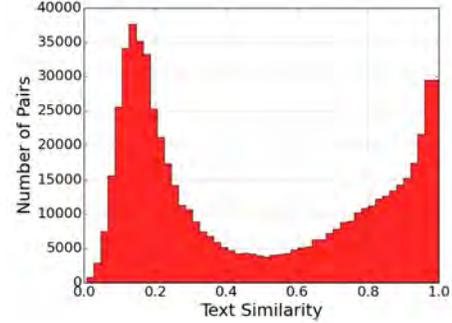


Fig. 3: Text Similarity for our Sampling Scheme. We use Jaccards Similarity between the ad unigrams as a measure of text similarity. The histogram shows that the sampling scheme results in both, a large number of near duplicates and non duplicates. Such a behaviour is desired to ensure a robust match function.

learnt by the classifier. It is interesting to note that the most informative features include, the spatial (Location), Temporal (Time Difference, Posting Date) and also the Linguistic (Number of Special Characters, Longest Common Substring) features. We also find that the domain specific features, extracted using regexs, prove to be informative.

TABLE I: Most Informative Features

Top 10 Features	
1	Location (State)
2	Number of Special Characters
3	Longest Common Substring
4	Number of Unique Tokens
5	Time Difference
6	If Posted on Same Day
7	Presence of Ethnicity
8	Presence of Rate
9	Presence of Restrictions
10	Presence of Names

The ROC curve for the classifier is presented in Figure 4. The classifier performs well, with an extremely low false positive rate. Such a behaviour is desirable for the classifier to act as a match function, in order to generate sensible results. High False Positive rates, increase the number of links between our records, leading to a ‘snowball effect’ which results in a break-down of the Entity Resolution process.

In order to minimise this breakdown, we need to heuristically learn an appropriate confidence value for our classifier. This is done by carrying out the ER process on 10,000 randomly selected records from our dataset. The value of size of the largest extracted connected component and the number of such connected components isolated is calculated

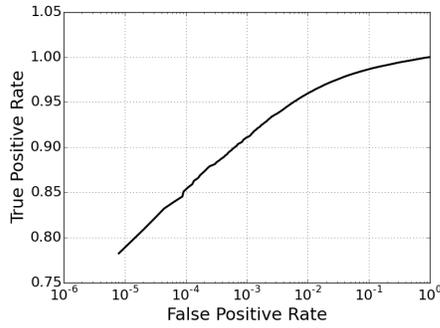


Fig. 4: ROC Curve for Match Function. (Log Scale) The ROC curve shows reasonably large True Positive rates for extremely low False Positive rates, which is a desirable behaviour of the match function.

for different confidence values of our classifier. This allows us to come up with a sensible heuristic for the confidence value.

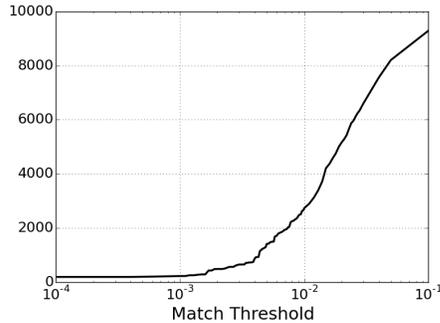


Fig. 5: The plot represents the size of largest connected component versus the value of the match threshold. Lower thresholds result in more granular clusters.

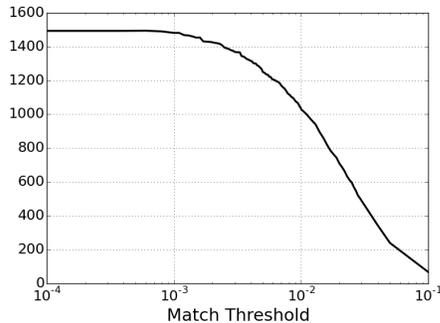
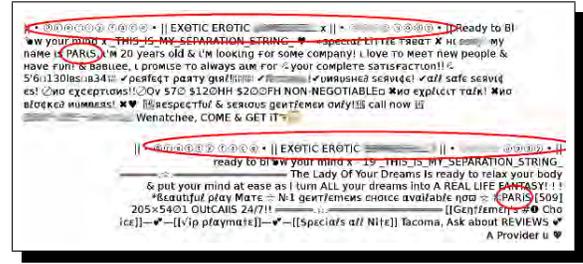


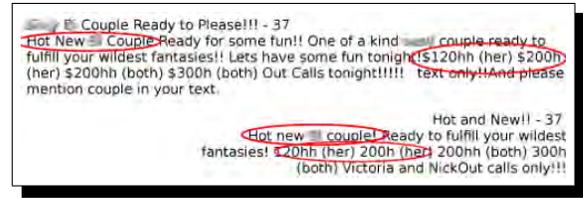
Fig. 6: The plot represents the number of connected components versus the match threshold. Lower thresholds result in a larger number of clusters.

D. Blocking Scheme

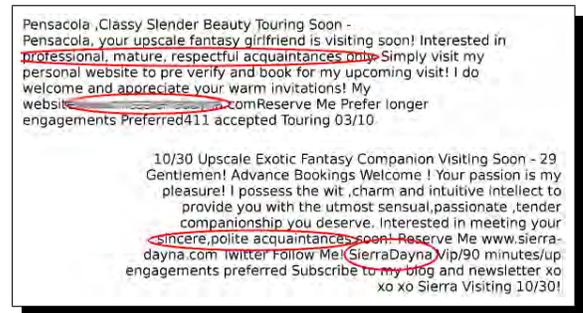
Our dataset consists of 5 million records. Naive pairwise comparisons across the dataset, makes this problem computationally intractable. In order to reduce the number of comparisons, we introduce a blocking scheme. The blocking scheme allows us to reduce the number of comparisons. We block the dataset on features like Rare Unigrams, Rare Bigrams and Rare Images. We then resolve the dataset across blocks.



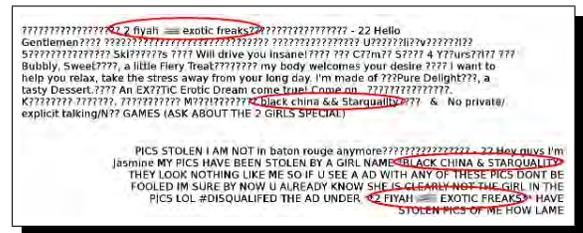
(a) This pair of ads have extremely similar textual content including use of non-latin and special characters. The ad also advertises the same individual, as strongly evidenced by the common alias, 'Paris'.



(b) The first ad here does not include any specific names of individuals. However, The strong textual similarity with the second ad and the same advertised cost, helps to match them and discover the individuals being advertised as 'Nick' and 'Victoria'.



(c) While this pair is not extremely similar in terms of language, however the existence of the rare alias 'SierraDayna' in both advertisements helps the classifier in matching them. This match can also easily be verified by the similar language structure of the pair.



(d) The first advertisement represents entities 'Black China' and 'Star Quality', while the second advertisement, reveals that the pictures used in the first advertisement are not original and belong to the author of the second ad. This example pair shows the robustness of our match function. It also reveals how complicated relationships between various ads can be.

Fig. 7: Representative results of advertisement pairs matched by our classifier. In all the four cases the advertisement pairs had no phone number information (strong feature) in order to detect connections. Note that sensitive elements have been intentionally obfuscated.

TABLE II: Results Of Rule Learning

Rule	Support	Ratio	Lift
$X_{minchars} \leq 250, 120000 < X_{maximgfrq}, 3 < X_{mnweeks} \leq 3.4, 4 < X_{mnmonths} \leq 6.5$	11	90.9%	2.67
$X_{minchars} \leq 250, 120000 < X_{maximgfrq} 4 < X_{mnmonths} \leq 6.5,$	16	81.25%	2.4
$X_{statesnorm} \leq 0.03, 3.6 < X_{uniquimgsnorm} \leq 5.2, 3.2 < X_{stdmonths}$	17	100.0%	2.5
$X_{statesnorm} \leq 0.03, 1.95 < X_{stdweeks} \leq 2.2, 3.2 < X_{stdmonths}$	19	94.74%	2.37



Fig. 8: Representative Entity isolated by our pipeline, believed to be involved in human trafficking. The nodes represent advertisements, while the edges represent links between advertisements. This entity has 802 nodes and 39,383 edges. This visualisation is generated using Gephi. [1]

III. RULE LEARNING

We extract clusters and identify records that are associated with human trafficking using domain knowledge from experts. We featurise the extracted components, using features like size of the cluster, the spatio-temporal characteristics, and the connectivity of the clusters. For our analysis, we consider only components with more than 300 advertisements. we then train a random forest to predict if the clusters is linked to human trafficking. In order to establish statistical significance, we compare the ROC results of our classifier in 4 cross validation for 100 random connected components versus the positive set. Table III lists the performance of the classifier in terms of False Positive and True Positive Rate while Table IV lists the most informative features for this classifier.

We then proceed to learn rules from our featureset. Some of the rules with corresponding Ratios and Lift are given in Table II. It can be observed that the features used by the rule learning to learn rules with maximum support and ratios, correspond to the ones labelled by the random forest as informative. This also serves as validation for the use of rule learning.

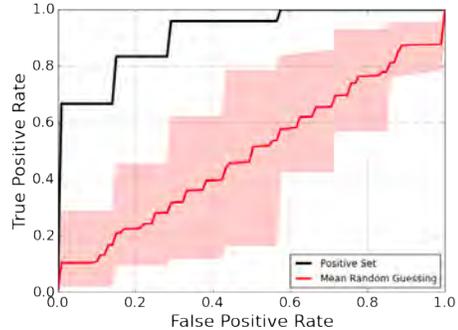


Fig. 9: ROC for the Connected Component classifier. The Black line is the ROC for the positive set, while the Red line is the average ROC for 100 randomly guessed predictors. The higher area under the curve for the positive test sets establishes statistical significance for our classifier.

TABLE III: Metrics for the Connected Component classifier

AUC	TPR@FPR=1%	FPR@TPR=50%
90.38%	66.6%	0.6%

TABLE IV: Most Informative Features

	Top 5 Features
1	Posting Months
2	Posting Weeks
3	Standard Deviation of Image Frequency
4	Normalised Number of Names
5	Normalised Number of Unique Images

IV. CONCLUSION

In this paper we approached the problem of isolating sources of human trafficking from online escort advertisements with a pairwise Entity Resoluition approach. We trained a classifier able to predict if two advertisements are from the same source using phone numbers as a strong feature, to generate training data for our classifier. The resultant classifier, proved to be robust, as evidenced from extremely low false positive rates. The DIG proposed by Szekely et. al. aims to build similar knowledge graphs using similarity score between each feature. This has some limitations. Firstly, we need labelled training data inorder to train match functions to detect ontological relations. The challenge is aggravated since this approach considers each feature independently making generation of enough labelled training data for training multiple match functions an extremely complicated task.

Since we utilise existing features as proxy evidence, our approach can generate a large amount of training data without the need of any human annotation. Our approach requires just learning a single function over the entire featureset, hence our classifier can learn multiple complicated relations between

features to predict a match, instead of the naive feature independence approach of DIG.

We then proceeded to use this classifier, in order to perform entity resolution using a heuristically learned value for the score of classifier, as the match threshold. The resultant connected components were again featurised, and a classifier model was fit before subjecting to rule learning. On comparison with [4], the connected component classifier performs a little better with higher values of the area under the ROC curve and the TPR@FPR=1% indicating a steeper, ROC curve. This may be due to the fact that we are able to generate larger amount of training data for our classifier, due to the entity resolution process. The learnt rules show high ratios and lift, for reasonably high supports as evidenced from Table II.

V. FUTURE WORK

While our blocking scheme performs well to reduce the number of comparisons, however since our approach involves naive pairwise comparisons, scalability is a significant challenge. One approach could be to design such a pipeline in a distributed environment. Another approach could be to use a computationally inexpensive technique to de-duplicate the dataset of the near duplicate ads, which would greatly help with regard to scalability.

In our approach, the ER process depends upon the heuristically learnt match threshold. Lower threshold values can significantly degrade the performance, with extremely large connected components. The possibility of treating this attribute as a learning task, would help making this approach more generic, and non domain specific.

Images associated with ads were also utilised as a feature for the match function. However, simple features like number of unique and common images etc., did not prove to be very informative. Further research is required in order to make better use of such features.

ACKNOWLEDGMENT

The authors would like to thank all staff at Carnegie Mellon, who made the Robotics Institute Summer Scholars program 2015 possible.

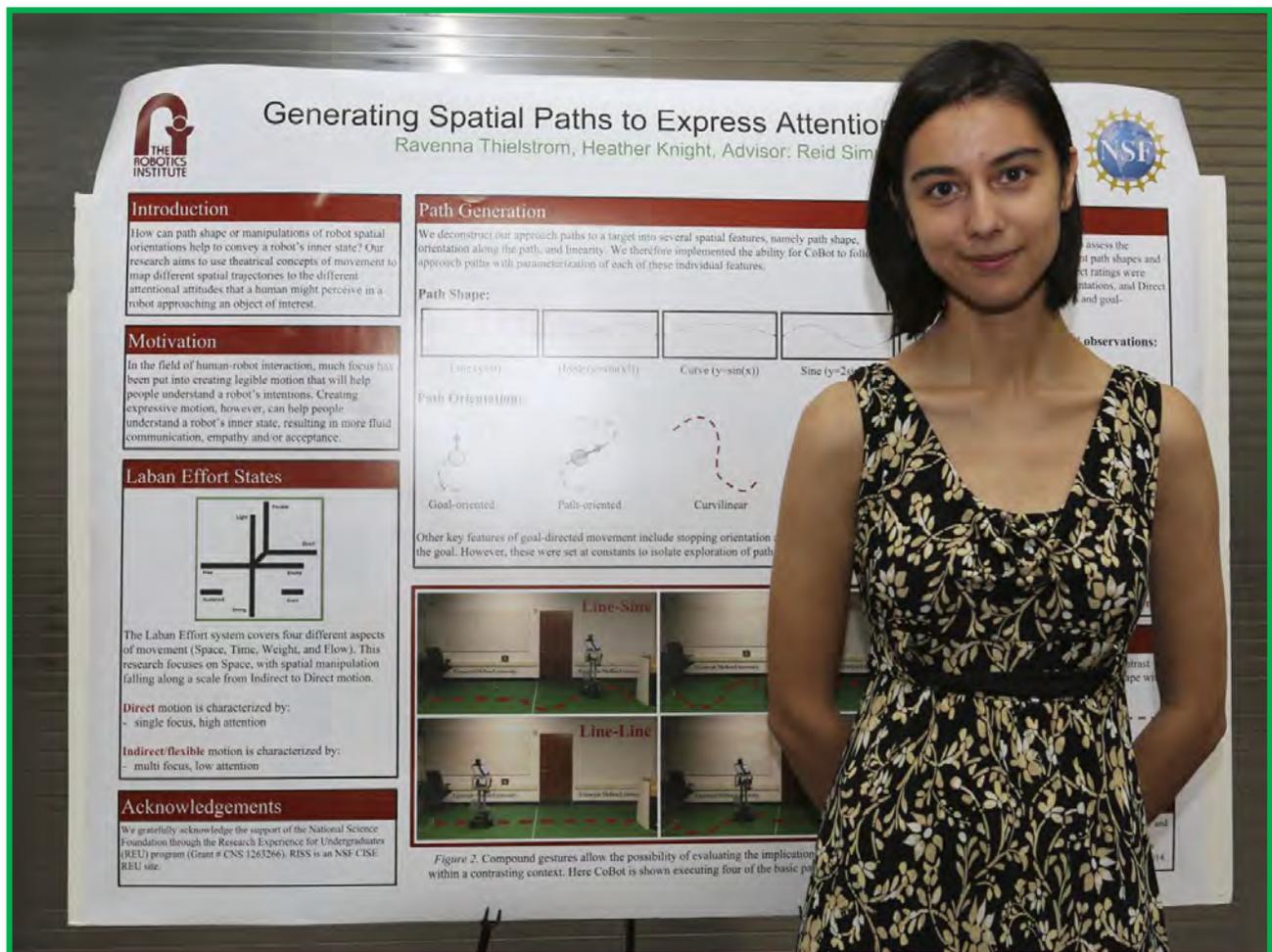
REFERENCES

- [1] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: An open source software for exploring and manipulating networks, 2009.
- [2] Omar Benjelloun, Hector Garcia-Molina, David Menestrina, Qi Su, Steven Euijong Whang, and Jennifer Widom. Swoosh: a generic approach to entity resolution. *The VLDB Journal/The International Journal on Very Large Data Bases*, 18(1):255–276, 2009.
- [3] Hamish Cunningham. Gate, a general architecture for text engineering. *Computers and the Humanities*, 36(2):223–254, 2002.
- [4] Artur Dubrawski, Kyle Miller, Matthew Barnes, Benedikt Boecking, and Emily Kennedy. Leveraging publicly available data to discern patterns of human-trafficking activity. *Journal of Human Trafficking*, 1(1):65–85, 2015.
- [5] Emily Kennedy. Predictive patterns of sex trafficking online. 2012.
- [6] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12:2825–2830, November 2011.

- [7] Pedro Szekely, Craig A. Knoblock, Jason Slepicka, Andrew Philpot, Amandeep Singh, Chengye Yin, Dipsy Kapoor, Prem Natarajan, Daniel Marcu, Kevin Knight, David Stallard, Subesware S. Karunamoorthy, Rajagopal Bojanapalli, Steven Minton, Brian Amanatullah, Todd Hughes, Mike Tamayo, David Flynt, Rachel Artiss, Shih-Fu Chang, Tao Chen, Gerald Hiebel, and Lidia Ferreira. Building and using a knowledge graph to combat human trafficking. In *Proceedings of the 14th International Semantic Web Conference (ISWC 2015)*, 2015.
- [8] Sriharsha Veeramachaneni and Ravi Kumar Kondadadi. Surrogate learning: from feature independence to semi-supervised classification. In *Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing*, pages 10–18. Association for Computational Linguistics, 2009.

Ravenna Thielstrom

RISS 2015



Generating Spatial Paths to Express Attentional Attitudes

Ravenna Thielstrom (Swarthmore College), Heather Knight and Reid Simmons (Carnegie Mellon Robotics Institute)

Abstract—Through observation of human body movement, we instinctively understand other humans’ attitudes and intentions, allowing us to interact with them more comfortably and productively. Robotic behavior, on the other hand, can seem hard to understand without direct communication, making interaction with robots less intuitive or comfortable. In the field of human-robot interaction, much focus has been put into creating legible motion that will help people understand a robot’s intentions. Creating expressive motion, however, can help people understand a robot’s attitudes, resulting in more empathy, acceptance, and fluid interaction. We used theatrical concepts of movement to map different characteristics of spatial trajectories to the different attentional attitudes that a human perceives in a robot. Approach paths to an object of interest were generated according to our deconstruction of a path into several combinable elements such as shape and orientation. A preliminary pilot was conducted to assess the directness of the different path characteristics, in which observers watched the robot execute different paths towards an object. Indirect ratings were highest for sine paths with path-orientations, and direct ratings were highest for linear paths and goal-orientations. Participants described the robot’s attitudes in similar ways for specific types of directness. Our pending study focuses on compound combinations of path segments to explore how successive contrasting paths might change how the robot’s attitudes are perceived. Overall, this research demonstrates the feasibility of using simple translations and rotations in space to convey humanlike attitudes in non-anthropomorphic robots with limited degrees of freedom.

I. INTRODUCTION

As mobile robots become more frequently used in daily life, the study of human-robot interaction becomes increasingly important in order to raise the ease and efficiency with which humans can collaborate and coexist with robots in the same environment. While a large part of human interaction stems from non-verbal cues in body language, the absence of such cues in robots means that humans can not as easily understand or feel comfortable with a robot’s actions. By replicating the types of motion that humans instinctively associate with specific mental states, we can generate familiar and easily interpreted behavior in robots. Such behavior specific to expressing the internal state of a robot, known as ‘expressive motion’, can convey the robot’s intentions and attitudes to humans without having to directly communicate with them. In addition, displaying humanlike movement, even in a distinctly unhumanlike robot, increases its acceptance by people in a social environment and the development of emotional bonds.

Our goal is not to copy human gestures, but rather to break down robot movement into aspects that are interpreted by humans as expressive of a humanlike inner state. This specifically focuses not on the body language of humanoid

Effort Vector	Fighting Polarity	Inducing Polarity
Space	Direct	Indirect
Time	Sudden	Sustained
Weight	Strong	Light
Flow	Bound	Free

TABLE I
THE LABAN EFFORT SYSTEM.

robots, but rather on the basics of movement: translation and rotation of the body of a non-anthropomorphic robot. The robot used in this study for this purpose is Carnegie Mellon’s CoBot, a mobile robot with an omnidirectional base with the ability to autonomously traverse Carnegie Mellon’s Computer Science building and complete travel-related tasks such as delivering a message from one office to another. Adding expressive motion to CoBot’s general autonomous movement would allow human observers to intuitively tell at a glance if CoBot is anxious to complete a current task, uncertain about its ability to complete a task, or idly wandering and available for task scheduling. A preliminary pilot was conducted to assess the perceived expressiveness of different path shapes and orientations towards an attentional goal, which were generated with the Indirect and Direct polarities of the Laban Space Effort in mind.

The generation of approach paths, and the methods by which we ran our pilot study will be detailed following a discussion of related work below. We found that the participants did observe CoBot’s behavior to align along Laban Effort vectors, thus confirming the usability of the Laban Effort system for robotic motion. In accordance with our hypothesis, Indirect ratings were highest for sine paths with path-orientations, and Direct ratings were highest for linear paths and goal-orientations. Additionally, participants consistently interpreted specific generated paths as expressive of specific attitudes. Finally, our preparation for a formal study will be discussed as well as our expectations of those pending results.

II. RELATED WORK

Humans have been shown to naturally anthropomorphize non-living animated objects based on their patterns of motion, as found in the well-known “Do Triangles Play Tricks?” study [1]. Based on this fact, past research in the area of robot motion has sought to define the parameters of motion which may result in perceived expressiveness, one such method being the categorization of movement according to the Laban Effort System.

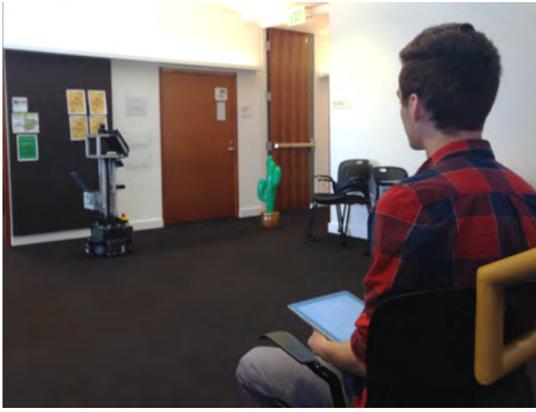


Fig. 1. A participant observes CoBot’s path towards its object of interest.

The Laban Effort System, created by R. Laban in 1963, is a means by which to discuss different types of motion, which is widely used in dance and theatre [4]. There are four main Laban Effort Vectors within this system, listed as Space, Time, Weight, and Flow, and motion can be categorized along a range between Fighting and Inducing polarities within each vector, listed in Table 1. This study concentrates only on Space and the respective polarities of Direct and Indirect along that vector. Direct movement is associated with a single focus and high attention, whereas Indirect movement is associated with multiple foci and low attention. The Laban Effort System is part of a larger overall method for studying motion known as Laban Movement Analysis, which has previously been successfully applied to movement studies with humanoid and flying robots [5] [6]. Our use of Laban Efforts builds off of previous work looking at the possibility of applying the system to simple mobile robots with low degrees of freedom [3].

This study also bears a similarity to research done on creating legible approach paths to objects of interest /citedragan. These paths were generated for a robotic arm to follow towards one of two adjacent objects on a table, and were evaluated for the legibility and predictability of the robot’s action (which of the objects it intended to pick up). Our study differentiates itself by focusing not on human ability to understand the robot’s intent, but rather on human ability to understand the robot’s expressed attitude towards its intended task.

III. APPROACH

Four different path shapes were generated, based on variations on a sinusoidal path (plus a completely straight line from the starting point to the goal). These were the shapes designated in fig:paths as Line, Hook, Curve, and Sine. CoBot’s omnidirectional base allows it to be oriented in a different direction than the direction it is moving in, so we also generated three different orientations for it to display on the path: along the path, towards the goal, or (not pictured on fig:angles) towards a point at a fixed offset from the goal.

Each participant was shown all possible combinations of the path features, and after each survey they were prompted

Path Shape:

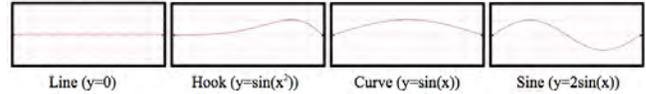


Fig. 2. Four different basic paths were conceived to test human perception of path shape.

Path Orientation:

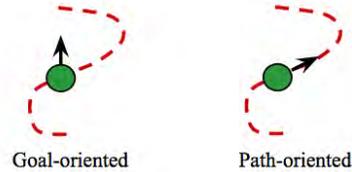


Fig. 3. Basic path orientations included in the experiment.

to fill out a survey on how well the path matched to direct and indirect attributes. Specifically, they were asked to rate on a five-point Likert scale how accurately each path corresponded to the following descriptors:

- pinpointed, single-focused, laser-like
- expansive, flexible, meandering
- deviating, avoiding, evading

Additionally, after viewing all the paths, each participant was interviewed about what interpretations they made about CoBot’s behavior, with questions such as:

- When did the robot seem most humanlike?
- When did the robot seem most artificial?
- When did the robot seem most or least attentive to the goal?
- Did you associate a narrative or storyline with any of the paths?
- Was the robot avoiding the goal at any point?
- Were there any points at which CoBot seemed like it was making decisions?
- Were there any paths that seemed unusual/inexplicable to you? Any paths which seemed exceptionally expressive?

These questions were designed to prompt the participant to discuss their observations, first impressions, and storytelling instincts regarding CoBot’s movement. Storytelling was encouraged in order to let the participants think about possible reasons for the CoBot to move in whatever path it was traveling.

IV. RESULTS

Seeing as most participants of the pilot study were Robotics Institute Summer Scholars, the data may not be representative of an ordinary sample of the human population. For example, a few participants noted during their interview that they may be biased towards thinking that irregular movement from CoBot is more likely to originate from programming error rather than emotions. Furthermore, as this was a pilot study, only 7 participants were used.

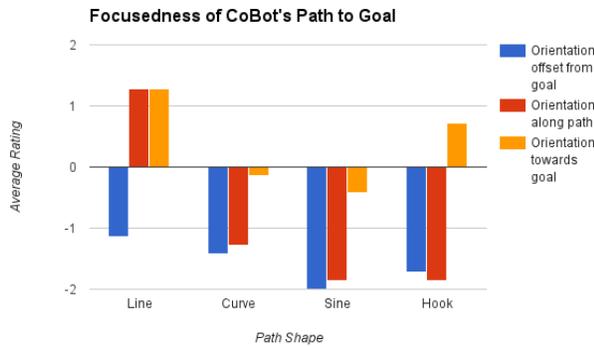


Fig. 4. Ratings of CoBot's focus

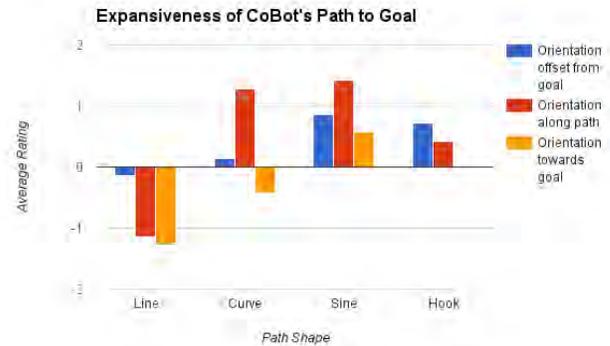


Fig. 5. Ratings of CoBot's expansiveness

The most significant result of our pilot study was the high consensus that the straight-line path for path and goal orientations (which is in fact the same path) deserved large positive ratings for focus and large negative ratings for both evasiveness and expansiveness. Thus the linear path corresponds as expected to the Laban conception of Direct movement. Assessments of this path as discussed by participants in the interview portion of the study included some impressions that CoBot seemed most humanlike and was paying the most attention to the goal on this path than on any other.

It should be noted that the linear path with an orientation slightly offset from the goal orientation did not receive the same ratings and in fact received ratings of the opposite sign for focus and evasiveness, along with the impressions that it was behaving artificially. A few participants stated they couldn't think conceive of a reason why CoBot would behave that way, "except as a joke, since it clearly knows where the goal is." Others associated the behavior with deception or avoidance. This contrast from demonstrates the importance of exact orientation in generating expressive motion— path shape alone is not enough to consistently convey an expressive state, and orientation must be taken into account.

Most nonlinear paths, in contrast to the non-offset linear paths, were rated as unfocused, evasive, and expansive. The sine curve obtained the most extreme ratings for both the focus question and the evasiveness question, being perceived as highly unfocused and highly expansive.

CoBot's behavior for paths along the sine curve were described more than once as "drunk", "confused", or "lost", and was generally determined to be the most artificial behavior.

Overall, the majority of participants agreed that orientation was the most important part of determining CoBot's attentional attitudes, and that looking at the goal made CoBot seem more attentive. One participant even mentioned, "When it wasn't looking at the goal, it looked less hurried" while another said moving while not looking at the goal made CoBot seem "aloof". While most participants had the impression that the linear path was the most humanlike and the sine curve the most artificial, one participant strongly

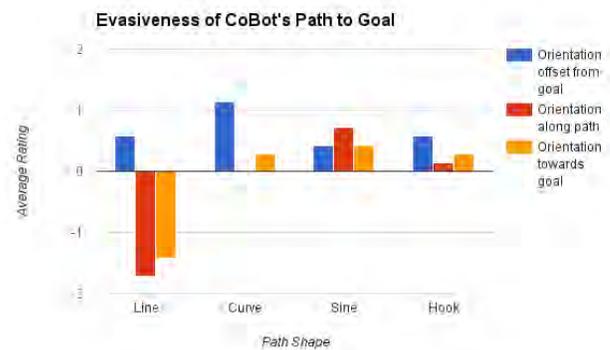


Fig. 6. Ratings of CoBot's evasiveness

expressed the opposite view with the reasoning that "Most humans look around, so the focused path felt more robotic."

The results of the survey demonstrated that the Laban Effort System is able to be applied to robotic motion as well as human motion, since our participants generally agreed upon ratings for each path in accordance with descriptors related to Laban Indirect and Direct categorization.

V. CONCLUSIONS

Humans observe other people's body language in order to get an understanding of their thoughts, feelings, and intentions without having to communicate directly with them. This makes human interaction more comfortable and cooperative. While efforts have been made to make human-robot interaction easier by copying human movement to humanoid robots, not all robots are humanoid and have the ability to replicate gestures specific to humanlike body parts. This research is therefore isolating simple x-y-theta movement for non- anthropomorphic robots in order to identify aspects of motion that are inherently expressive. We're using concepts of motion from theatre to evaluate robot movement into identifiable characteristics, starting with the Indirect-Direct scale of movement in the domain of Space, with the goal of being able to map these characteristics in a path to specific types of attentional attitudes. To determine these characteristics we generated varying approach paths to an

object of interest, according to our deconstruction of a path into several combinable elements, such as shape and orientation. Participants described the robot's attitudes in similar ways for specific types of directness, and were more likely to engage in storytelling when describing the robot's inner state if the path was more indirect.

Our pending study focuses on compound combinations of path segments to explore how successive contrasting paths might change how the robot's attitudes are perceived. Overall, this research demonstrates the feasibility of using simple translations and rotations in space to convey humanlike attitudes in non-anthropomorphic robots with limited degrees of freedom.

ACKNOWLEDGMENT

We gratefully acknowledge the support of the National Science Foundation through the Research Experience for Undergraduates (REU) program (Grant # CNS 1263266). RISS is an NSF CISE REU site.

REFERENCES

- [1] F. Abella, F. Happb and U. Fritha. Do triangles play tricks? Attribution of mental states to animated shapes in normal and abnormal development. *Cognitive Development*, Vol. 15:1, pp. 116, JanuaryMarch 2000.
- [2] A.D. Dragan, K.T. Lee, and S.S. Srinivasa, "Legibility and Predictability of Robot Motion". International Conference on Human-Robot Interaction (HRI), 2013.
- [3] H. Knight and R. Simmons, "Expressive Motion with X, Y and Theta: Laban Effort Features for Mobile Robots," in Proceedings of International Symposium Robot and Human Interactive Communication, Edinburgh Scot- land, August 2014.
- [4] R. Laban. *Modern Educational Dance*. Macdonald & Evans, 1963
- [5] M. Masuda, S. Kato and H. Itoh. Laban-Based Motion Rendering for Emotional Expression of Human Form Robots. Chapter in *Knowledge Management and Acquisition for Smart Systems and Services Lecture Notes in Computer Science*, Volume 6232: 49-60, 2010.
- [6] M. Sharma, et al. Communicating affect via flight path: exploring use of the laban effort system for designing affective locomotion paths. In *Proceedings Int'l Conference on Human-Robot Interaction*, 2013

Hadi Salman (1/2)

RISS 2015

A Novel Skidding Model for a Nonholonomic System

Hadi Salman, Tony Dear (PHD)
Advisor: Howie Choset

Carnegie Mellon University
The Robotics Institute

Introduction

The skateboard is a simple system with rich behavior. Many models usually assume nonholonomic constraints to represent two-wheeled wheels.

We consider changing these constraints to analyze skidding behavior.



Common skidding models have been used in the modeling of nonholonomic mechanical systems.

- 1. Relaxing the nonholonomic constraints [1]
- 2. Relaxing the nonholonomic constraints [1]
- 3. Using Rayleigh dissipative functions [2]

Inspired by how Sidiq & Sarkar modeled skidding for a unicycle nonholonomic mobile robot [4], we present a novel skateboard model that accounts for skidding by adding an extra variable to the configuration space, of the skateboard, and relates the force that actuates this variable to the skidding angle using Pacejka's magic formula [3].

Rayleigh Model

- Replaces the "hard" nonholonomic constraints with "soft" dissipative forces.
- Uses an "energy-like" Rayleigh Dissipative Function

Directly related formula:

$$R = \frac{1}{2} k_v \dot{\phi}^2$$

Single parameter $\rightarrow \frac{\partial}{\partial t}$ \rightarrow Soft Dissipative Forces along the axis of the Wheelset

Discussion

- The new model and the Rayleigh model are very closely.
- Advantages of new model:
 - Allows for more freedom to account physical conditions
 - Experimentally verified skidding (Pacejka's formula is obtained from experimental data fittings)
- Comparison and correlation with Rayleigh model:
 - Rayleigh model is unable to reproduce simulation results for certain values of the load, no matter what the camber angle is. The vertical extended flexibility of Pacejka's parameter.
 - On the other hand, Rayleigh model is directly to real conditions.

Mechanical Model

Generalized Coordinates:



- The rotor and the wheelset's angles are actuated.
- Generalized Coordinates: $q = [x, y, \theta, \phi, \psi, \beta]$
- Euler-Lagrange dynamics formulation: $\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} = \tau + \lambda^T (A - T)$

$L(q, \dot{q})$	Kinetic Energy $T(q, \dot{q})$
$\frac{1}{2} m (\dot{x}^2 + \dot{y}^2) + \frac{1}{2} I \dot{\theta}^2$	Generalized forces (inputs)
$\frac{1}{2} k_v \dot{\phi}^2$	Lagrange multipliers
$\lambda^T [A - T]$	Constraint reaction forces that arise due to the nonholonomic constraints due to skidding.

- Constraint: $A\dot{q} = 0$
- $A =$

$$\begin{bmatrix} -\sin(\theta + \phi) & \cos(\theta + \phi) & l \cos(\delta) & 0 & 0 & l \cos(\delta) \\ -\sin(\theta - \phi) & \cos(\theta - \phi) & -l \cos(\delta) & 0 & 0 & -l \cos(\delta) \end{bmatrix}$$
- Assumption: $F_x = F_y = F$, i.e. equal skidding on the back and front wheelsets

Pacejka's Formula

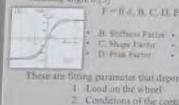
Relates the lateral traction force F on wheel to the skidding angle δ [3]:

$$F = f_0 D E S \sigma$$

- B: Softness Factor $\rightarrow S_0$, Vertical Shift
- C: Shape Factor $\rightarrow S_1$, Horizontal Shift
- D: Peak Factor $\rightarrow S_2$, Curvature factor

These are fitting parameter that depend on:

- Load on the wheel
- Conditions of the contact surface
- Camber angle of the wheel



Can start simulation!

Simulation

Dry Asphalt Road: $\phi(t) = 0.3 \sin(t)$ / $\psi(t) = 0.5 \sin(t)$ / Load = 2.54 N
Camber angle = 0.52 rad / duration: 35 sec



Dry Asphalt Road: $\phi(t) = 0.3 \sin(t)$ / $\psi(t) = 0.5 \sin(t)$ / Load = 5.08 N
Camber angle = 0.3 rad / duration: 35 sec



Dry Asphalt Road: $\phi(t) = 0.2$ / $\psi(t) = 0.5 \cos(t)$ / Load = 5.08 N /
Camber angle = 0.3 rad / duration: 35 sec



A Novel Skidding Model for a Snakeboard

Hadi Salman, Tony Dear, and Howie Choset

Abstract—The Snakeboard is a well-studied example for mechanical systems analysis, largely because of its simultaneous richness in behavior and simplicity in design. However, few Snakeboard models incorporate skidding as a violation of the rigid nonholonomic constraints. In this paper, we present a novel Snakeboard model that accounts for skidding by adding an extra variable to the configuration space of the Snakeboard. It relates the force that actuates this variable to the skidding angle using Pacejka’s magic formula, an experimentally verified model that relates the traction forces on the wheel to the skidding angle. We compare our model to the Rayleigh dissipation function model and present some correlation between the parameters of the two models. This helps figuring out what physical meaning the simple Rayleigh dissipation function model’s coefficient signifies.

Keywords—Nonholonomic Constraints, Dynamics, Kinematics, Skidding.

I. INTRODUCTION

THE Snakeboard is a canonical example of a mixed nonholonomic mechanical system, one whose motion is governed by both kinematic constraints and dynamics [1], [2]. The Snakeboard shown in Fig. 1 consists of two sets of wheels that can rotate about the center point of the axle. To ride the Snakeboard, one alternates between rotating one’s torso with one’s ankles to move the wheelsets. The mechanical model (Fig. 2) has a rotor situated at the center of the longitudinal axis to simulate the human rider’s torso. The rotor and wheel axle angles are actuated, and the latter are nonholonomically constrained, allowing the Snakeboard to locomote due to input joint actuation.

Skidding is a common phenomenon that occurs when the wheels of a system slip sideways, i.e. the nonholonomic constraints imposed on a wheel



Fig. 1. A Snakeboard, composed of a rigid axis and two footrests on rotating wheelsets.

are violated. In an idealized mechanical system, models often ignore the effects of skidding, whereas in actual physical systems, the effects of skidding can be considerable, especially for systems such as a Snakeboard whose motions rely basically on rotating the axes of the wheels.

Early approaches in modeling skidding used corrective control strategies by the aid of sensor measurements[3]–[5]. Different approaches were then developed in order to include skidding in the dynamics of the system, and to eliminate any sort of dependency on GPS or measurements from sensors in order to quantize the skidding that is happening and control the system accordingly.

Sidek and Sarkar [6] included skidding in the dynamics of a 2-wheeled mobile robot by relaxing the nonholonomic constraints, adding an additional linear variable to the generalized coordinates of the system and relating the skid angle to the lateral traction force on the wheel that arises due to skidding. Bazzi et al. [7] then proposed two other methods that include the skidding effects in the kinematic model of vertical rolling disk; one which relates skidding to the Lagrange multipliers of the kinematic constraints, and another which relates skidding to the generalized velocities of the actuated degrees of freedom of the vertical rolling disk.

Dear et al. [8] then proposed a novel skidding model that is based on the addition of Rayleigh dissipation functions, and it was the first time skidding effects have been incorporated into a mixed system such as a Snakeboard.

H. Salman is with the Department of Mechanical Engineering, American University of Beirut, Beirut, Lebanon e-mail: hcs03@mail.aub.edu

T. Dear and H. Choset are with the Robotics Institute at Carnegie Mellon University, Pittsburgh, PA, 15213, USA. emails: tony-dear@cmu.edu, choset@cs.cmu.edu

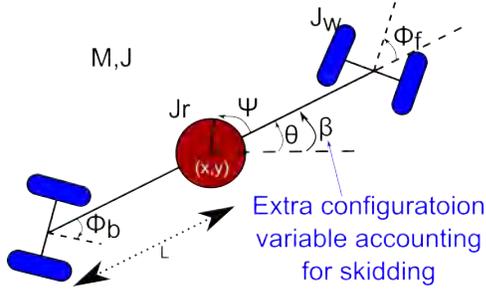


Fig. 2. The configuration variables of the Snakeboard. We will assume in this paper that the angles the wheelset sets make with the axis of the snakeboard are coupled such that $\phi_f = \phi_b = \phi$.

Based on what Sidek and Sarkar [8] did to model skidding for a 2-wheeled mobile robot, we present in this paper a novel method to model skidding for a more complex and hybrid system: the Snakeboard. We show how we can account for skidding in the dynamics of the Snakeboard by adding an extra "rotational" variable to the configuration space and relating the lateral dissipative force, that actuates this variable, to the skidding angle through Pacejka's magic formula. We simulate this new model under different conditions and compare the results with Rayleigh dissipation function model. Finally, we discuss some correlations between the parameters of the two models. This helps us find out what the Rayleigh model's coefficient physically signifies, and what conditions that affect skidding does it capture.

II. MOTIVATION OF THE NEW MODEL

In this section, we present some background material on the nature of skidding, and we show how skidding appears in a nonholonomic system, such as the Snakeboard, and how it gives the inspiration for the new model.

A. Wheel Skidding

Skidding occurs due to the lateral, cornering force, generated when negotiating a turn [7]. Due to skidding, the wheel traverses a path along a direction that is deviated from the direction in which the wheel is oriented in the ideal case. This is all illustrated in Fig.3.

B. Skidding in a Snakeboard

For a Snakeboard, skidding can be noticed when the nonholonomic constraints along the wheelsets'

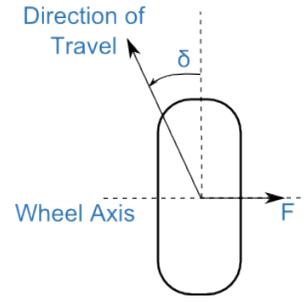


Fig. 3. The lateral force F on the wheel due to skidding. Notice the deviation of the direction of travel from the ideal direction by a skid angle δ .

axes are violated. This leads to a velocity v_1 that relaxes the nonholonomic constraint along the wheelset axes as shown in Fig. 4. This happens at each of the two wheelsets but in opposite directions due to the coupling of the rotation of the front and back wheelsets which will be discussed in the next section and will be assumed hereafter. This leads to a net rotation of the whole system. It is then meaningful to add an extra variable to the configuration space of the Snakeboard which captures the rotation that is caused by skidding.

C. Rayleigh Dissipation Function Model

We will use this Rayleigh dissipation function model in order to verify the results of the simulation. We present here a brief overview of this model that was proposed by Dear et al.[8] This model is based on replacing the "hard" nonholonomic constraints with soft dissipative forces modeled by a skidding Rayleigh dissipation function.

$$R_{skid}(q) = \frac{1}{2}k_s v^2 \quad (1)$$

where k_s is the skidding coefficient, and v is the velocity in the skid direction.

This dissipative function is then augmented into the Euler-Lagrange equations in order to derive the equations of motion of the system.

III. MATHEMATICAL MODEL

The Snakeboard is a multi-bodied mechanical system with two nonholonomic constraints acting on its wheel sets. We derive the equations of motion of the Snakeboard using the Lagrangian approach. Similar to what Sidek and Sarkar [6] did for a 2-wheeled nonholonomic mobile robot, we

added an extra variable β to the vector of generalized coordinates of the Snakeboard which becomes $q = \{x, y, \theta, \psi, \phi, \beta\}$ where x, y and θ denote the position and orientation of the Snakeboard in the inertial coordinate frame while ψ and ϕ denote the rotor angle and the coupled angle of the wheelsets respectively as shown in Fig. 3. The new variable β denotes an extra rotational angle of the whole body of the Snakeboard due to skidding.

The actuated variables are the rotor angle ψ , in addition to the coupled wheelset angle ϕ . The mass of the whole Snakeboard is M centered in the middle of the rotor. J, J_r , and J_w are the moments of inertia of the Snakeboard, the rotor, and the wheelsets respectively. We will assume as in [8] that $ML^2 = J + J_r + 2J_w$ is the total inertia of the system, where the total length of the Snakeboard is $2L$.

A. Equations of Motion

We use the Euler-Lagrange dynamics formulation to derive the equations of motion of our modified system. The general form of these equations is

$$\frac{\partial}{\partial t} \frac{\partial L(q, \dot{q})}{\partial \dot{q}} - \frac{\partial L(q, \dot{q})}{\partial q} = \tau + A(q)^T (\lambda - T) \quad (2)$$

where $L(q, \dot{q})$ is the Lagrangian of the system, $\tau = \{0, 0, 0, \tau_r, \tau_w, 0\}$ is the vector of generalized forces (input torques), $A(q)$ is a 2×6 matrix representing the nonholonomic constraints satisfying $A(q)\dot{q} = 0$, $\lambda \in \mathbb{R}^2$ is the vector of Lagrange multipliers, and $T = \{F_1, F_2\}$ represents dissipative traction forces that relax the nonholonomic constraints due to skidding. We will assume that the skidding that occurs on the front and back wheelsets is the same i.e. $F_1 = F_2 = F$.

The Lagrangian of the Snakeboard is simply the kinetic energy of the system and is given by $L(q, \dot{q}) = \frac{1}{2} \dot{q}^T D \dot{q}$, where D is the mass matrix associated with our system and is given by

$$\begin{pmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & J + J_r + 2J_w & J_r & 0 & J + J_r + 2J_w \\ 0 & 0 & J_r & J_r & 0 & J_r \\ 0 & 0 & 0 & 0 & 2J_w & 0 \\ 0 & 0 & J + J_r + 2J_w & J_r & 0 & J + J_r + 2J_w \end{pmatrix} \quad (3)$$

The nonholonomic constraints satisfying $A(q)\dot{q} = 0$ matrix is represented by the matrix $A(q)$ given by

$$\begin{pmatrix} -\sin(\theta + \phi) & \cos(\theta + \phi) & L \cos(\phi) & 0 & 0 & -L \cos(\phi) \\ -\sin(\theta - \phi) & \cos(\theta - \phi) & -L \cos(\phi) & 0 & 0 & L \cos(\phi) \end{pmatrix}$$

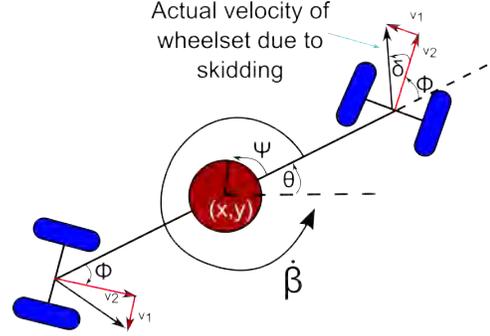


Fig. 4. Skidding velocities in a Snakeboard. Notice that the two linear velocities v_1 along the 2 wheelsets lead to a net angular velocity β of the snakeboard.

The constraints equations along with the Euler-Lagrange equations make up 8 equations, but we have nine variables: six configuration variables, the lateral dissipative traction force F , and the two lagrange multipliers associated with the two non-holonomic constraints. Thus in order to render our system of equations solvable, we will eliminate the traction force F by relating it to the skidding angle.

B. Traction Force Model

In order to model the lateral traction force on the wheels, we will use what Sidek and Sarker [6] used in their 2-wheeled mobile robot model: Pacejka's Magic formula [9]. This is an elegant formula based on curve fitting and commonly used when modeling tires. It relates the lateral force on a wheel to the angle of skid in the following manner:

$$F = D \sin(C \arctan(B\eta)) + \Delta S_v \quad (4)$$

where

$$\eta = (1 - E)(\delta + \Delta S_h) + \frac{E}{B} \arctan(B(\delta + \Delta S_h))$$

δ is the skidding angle and the coefficients $B, C, D, E, \Delta S_h, \Delta S_v$ are fitting parameters that are dependent on the following factors:

- The load on the wheel
- The conditions of the surface of contact
- The camber angle of the wheel

The relations between the above coefficients and the 3 factors can be found in the literature [9].

C. Relating Traction Force to Configuration variables in a Snakeboard

We will assume for a Snakeboard that the distance that separates the two wheels of the same wheelset

is very small so that we can model the wheelset as a single wheel that is allowed to roll and is actuated by rotating it around a vertical axis perpendicular to the X-Y plane. This allows us to express the lateral traction force at the center of the wheelset to the skidding angle δ as shown in Fig. 4.

We now relate the skid angle to the variables of our system. The skidding angle δ can be defined, as shown in Fig. 4, as

$$\delta = \arctan\left(\frac{v_1}{v_2}\right) \quad (5)$$

where v_1 is the velocity due to skidding along the wheelset axis, and v_2 is the velocity of the wheelset perpendicular to its axis (its velocity in the ideal case). These two velocities can be related to the configuration velocities and variables through

$$v_1 = L\dot{\beta} \cos \phi \quad (6)$$

$$v_2 = \dot{x} \cos(\theta + \phi) + \dot{y} \sin(\theta + \phi) + L(\dot{\beta} + \dot{\theta}) \cos(\phi) \quad (7)$$

Now we have related the lateral traction force on the wheels to the configuration variables and velocities, and we end up with eight equations in eight unknowns system which can be solved numerically.

IV. SIMULATION AND RESULTS

In order to analyze the new dynamics model that is presented by our model, we will simulate its response under certain inputs and compare it with the Rayleigh dissipation function model.

We numerically solve the equations of motion where the torques of the actuated degrees of freedom are equated to the following PD controller

$$\tau_r = K_p(\psi_d - \psi) + K_d(\dot{\psi}_d - \dot{\psi}) \quad (8)$$

$$\tau_w = K_p(\phi_d - \phi) + K_d(\dot{\phi}_d - \dot{\phi}) \quad (9)$$

where τ is the generalized torque input, ψ_d and ϕ_d are the desired trajectories of the controlled variables, and K_p and K_d are the proportional and derivative gains of the controller respectively. Throughout the simulations, we keep the following parameters constant: $J_r = 4$, $J_w = 1$, $L = 4$, $K_d = 25$, and $K_p = 4.5$.

We will vary the mass M of the Snakeboard in order to study the effect of changing the load on the

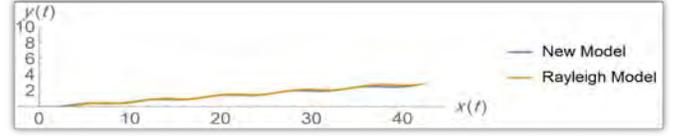


Fig. 5. The trajectory followed by the Snakeboard for two different skidding models on a dry asphalt road. Gait1, Load = 2.54N, Camber angle = 0.52rad, duration: 35sec

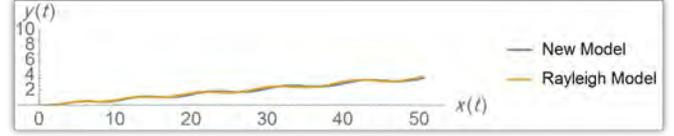


Fig. 6. The trajectory followed by the Snakeboard for two different skidding models on a dry asphalt road. Gait1, Load = 5.08N, Camber angle = 0.3rad, duration: 35sec

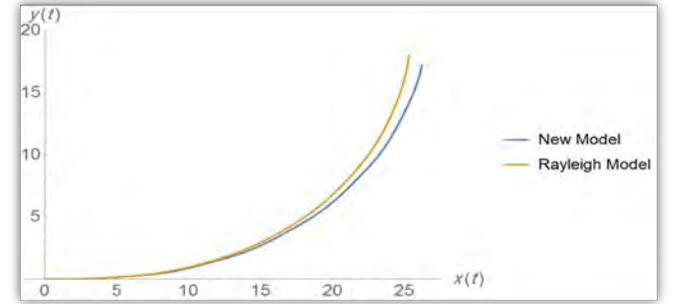


Fig. 7. The trajectory followed by the Snakeboard for two different skidding models on a dry asphalt road. Gait2, Load = 5.08N, Camber angle = 0.3rad, duration: 35sec

wheels on skidding. J is changed as well using the relation

$$J = ML^2 - J_r - 2J_w \quad (10)$$

Three different simulations are conducted; Fig. 5 shows the trajectory followed by the Snakeboard, subjected to gait 1, on a dry asphalt road with a load of 2.54 N (the weight of the Snakeboard) and a camber angle of 0.52 rad using the new skidding model and Rayleigh dissipation function model. Fig. 6 shows the results of another simulation where we kept the same gait as the simulation 1 but changed the load (5.08 N) and the camber angle (0.3 rad) of the wheels. Fig. 7 shows the trajectory followed by the Snakeboard under the same conditions of simulation 2 but with changing the gait.

The results of the three simulations shows a great agreement between the new model and the Rayleigh dissipation function model.

TABLE I
TABLE OF INPUTS USED FOR EACH GAIT

Gait	$\phi(t)$	$\psi(t)$
1	$0.3\sin(t)$	$0.5\sin(t)$
2	0.2	$0.5\cos(t)$

V. DISCUSSION

It is evident from the simulation results that the new skidding model for the Snakeboard successfully accounts for skidding as it gives almost the same behaviour as the Rayleigh dissipation function model. The advantages of this new model is that the coefficients of Pacejka’s magic formula, the key relationship of this model, are empirical. Thus for each condition of the system and the environment around the system, the coefficients that reflect these conditions can be experimentally derived. [9]

Another advantage is that this new model allows for more freedom to account for different physical conditions since the coefficients we mentioned above are sensitive to various physical conditions, such as the load on the wheels and the conditions of the ground in addition to the camber angle of the wheels.

A. Correlations between the new model and Rayleigh dissipation function model

In an attempt to correlate the skidding coefficient of the Rayleigh function k_s with the parameters used in this new model, we varied the parameters of the new model and get some results for the simulation, then we tried to find the value of k_s which gives the same response when using Rayleigh dissipation function model. This allows us to figure out what the k_s coefficient physically reflect. We found out that Rayleigh model is unable to reproduce simulation results for certain values of the load, no matter what the camber angle is. This confirms the extended flexibility of Pacejka’s parameters.

On the other hand, we noticed that by fixing all the parameters of the new model and changing only the D parameter, we were able to reproduce the same results using the Rayleigh model by varying the value of K_s . This signifies that Rayleigh model corresponds directly to road conditions since the only justification of the constant values of the other parameters that remained fixed during these testings

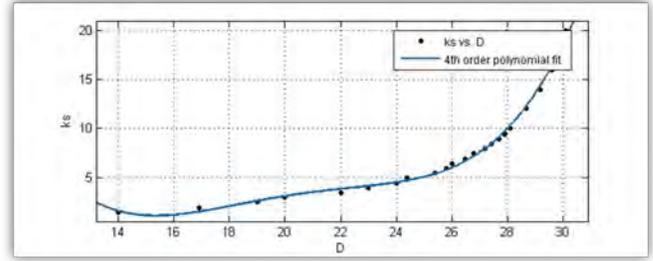


Fig. 8. 4th order polynomial curve fitting for the k_s as a function of the parameter D of the new model.

is the change in the coefficients inside these parameters [9] which correspond to changes in the road conditions.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we analyzed the effects of skidding on a Snakeboard. Based on a previous simple model for modeling skidding for a 2-wheeled mobile robot[Sidek cite], we presented a novel skidding model that accounts for skidding in a mixed system by adding a rotational variable to the configuration space and incorporating lateral traction dissipative forces into the equations of motion using Pacejka’s magic formula. We simulated the system under different conditions and for different gaits and compared the results to those of the Rayleigh dissipation function model. Finally, we discussed how the two model correlate and compare to each other.

In future work, we envision investigating more the correlations between the Rayleigh model parameter and the parameters of Pacejka’s magic formula. We will also analyze motion planning for Snakeboards incorporating this model.

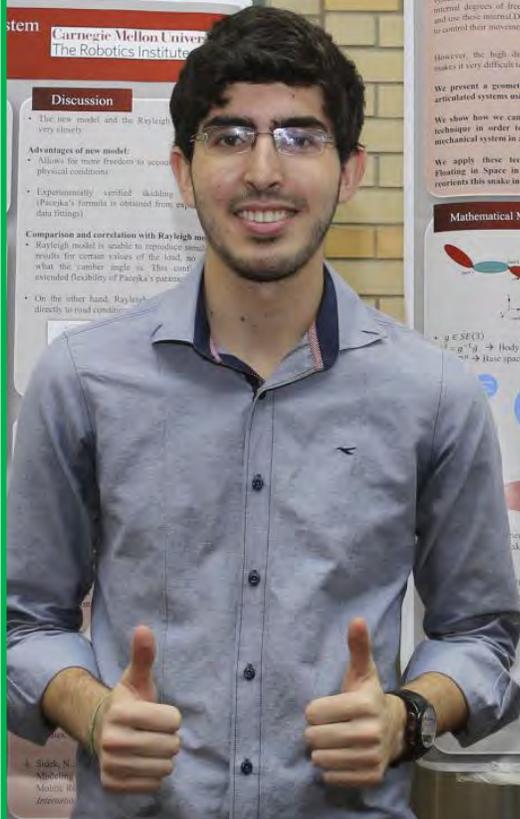
REFERENCES

- [1] J. Ostrowski, J. Burdick, A. Lewis, and R. Murray, “The mechanics of undulatory locomotion: The mixed kinematic and dynamic case,” *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, vol. 2, pp. 1945–1951, 1995.
- [2] E. Shamma, H. Choset, and A. A. Rizzi, “Towards a unified approach to motion planning for dynamic underactuated mechanical systems with non-holonomic constraints,” *The International Journal of Robotics Research*, vol. 26, no. 10, pp. 1075–1124, 2007.

- [3] C. B. Low and D. Wang, "Integrated estimation for wheeled mobile robot posture, velocities and wheel skidding and slipping perturbations," *Proceedings of the IEEE/ICRA International Conference on Robotics and Automation*, pp. 2355–2360, 2007.
- [4] —, "Gps-based path following control for a car-like wheeled mobile robot with skidding and slipping," *IEEE Trans. Control Syst. Technol.*, pp. 340–347, 2008.
- [5] —, "Gps-based tracking control for a car-like wheeled mobile robot with skidding and slipping mechatronics," *IEEE/ASME Trans.*, pp. 480–484, 2008.
- [6] N. Sidek and N. Sarkar, "Dynamic modeling and control of nonholonomic mobile robot with lateral slip," *Third International Conference on Systems*, pp. 35–40, 2008.
- [7] S. Bazzi, E. Shamma, and D. Asmar, "A novel method for modeling skidding for systems with nonholonomic constraints," *Nonlinear dynamics*, vol. 76, no. 2, pp. 1517–1528, 2014.
- [8] T. Dear, S. D. Kelly, M. Travers, and H. Choset, "Snakeboard motion planning with viscous friction and skidding," *International Conference on Robotics and Automation*, 2015.
- [9] P. H. B, E. Bakker, and L. Nyborg, "Tyre modeling for use in vehicle dynamics studies," *Delft University of Technology*, pp. 35–40, 2008.

Hadi Salman (2/2)

RISS 2015



Shape Based Geometric Motion Planning for an Underactuated Highly-Articulated System

Hadi Salman, Chuohui Gong (PHD)
Advisor: Howie Choset

Carnegie Mellon University
The Robotics Institute

Introduction

Mechanical systems that use their internal shape changes to control their movements have always possessed the geometric control community.

Snake robots are such systems that have many internal degrees of freedom and use these internal DOF's to control their movements.

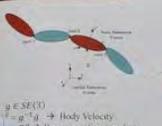
However, the high-dimensionality of these systems makes it very difficult to control them.

We present a geometric solution to control highly articulated systems using "Shape Basis".

We show how we can benefit from the shape basis technique in order to generate gaits that move a mechanical system in a desired direction.

We apply these techniques to a Snake Robot Floating in Space in order to generate gaits that reorients this snake in any direction.

Mathematical Model



$g \in SE(3)$
 $\dot{g} = g \cdot \xi \rightarrow$ Body Velocity
 $\xi \rightarrow$ Base space (joint angles)

Control Matrix i.e. relates the body velocity to the joint velocities

Change in orientation rate ξ with time

$\dot{r} = \int \text{Curl} A(r) dr$

Shape Basis

$r(t) = (r_1(t), \dots, r_n(t)) \in \mathbb{R}^n$

Controlled by θ (underactuated) \Rightarrow
 $r_1(t) = \sin(\theta_1) \cos(\theta_2)$
 $r_2(t) = \sin(\theta_1) \sin(\theta_2)$

Shape Basis
 $\sigma = \begin{pmatrix} \sigma_1 \\ \sigma_2 \end{pmatrix} \in \mathbb{R}^2$

$\xi = A(r) \frac{\partial r}{\partial \sigma} \dot{\sigma}$

Example:

- $\theta_1(t) = \text{angle of the } e^{\theta}$ point
- $\theta_2(t) = \pi/3$
- Shape Basis: $(\sin(2t), \cos(2t))$
- $\sigma_1(t) = 9 \cos(0.5 \sin(2t))$
- $\sigma_2(t) = 9 \sin(0.5 \sin(2t))$

Simulation

- θ_1 : angle of the e^{θ} axis of the 3x2 matrix $A(r)$
- θ_2 : angle of rotation around inertial z axis
- β : angle of rotation around inertial y axis
- γ : angle of rotation around inertial x axis

→ HEBART Function

Around X axis	Around Y axis	Around Z axis

Conclusion

- Shape Basis reduces the dimensionality of the controlling Base space
- Using shape bases, we are able to easily generate gaits that reorient the Floating Snake Robot in any desired direction simply by looking at the three height function associated with our robot.

Future Work

- Optimizing the paths for controlling orientation of a Floating Snake using Minimum Perturbation Technique
- Optimization the choice of a shape basis
- Exploring more mechanical systems that can be controlled using shape bases

References

- Murray, R. M., Li, Z., & Sastry, S. S. (1994). *A Mathematical Introduction to Robot Manipulation*. CRC Press.
- Travers, M., Hanson, R., & Choset, H. (2011). Minimum Perturbation Coordination of SDR's. American Control Conference, Washington DC.
- Shimizu, E., Choset, H., & Rizzi, A. (2007). Geometric Motion Planning for Two Classes of Underactuated Mechanical Systems. *The International Journal of Robotics Research*, 19(4), 104-107.

Shape Based Geometric Motion Planning for an Underactuated Highly-Articulated System

Hadi Salman, Chaohui Gong, and Howie Choset

Abstract—Mechanical systems that use their internal shape changes to control their movements have always interested the geometric control community. Snake robots are such systems that have many internal degrees of freedom and use these internal degrees of freedom to control their movements. However, the high dimensionality of these systems makes it very difficult to control them. We present a geometric solution to control highly articulated systems using “Shape Basis”. We show how we can benefit from the Shape Basis technique in order to generate gaits that move a mechanical system in a desired direction. We apply these techniques to a snake robot floating in space in order to generate gaits that reorient this snake in any direction.

Index Terms—Geometric Mechanics, Geometric Control, Lie Group.

I. INTRODUCTION

THE geometric mechanics community have always showed interest in studying flying inertial systems that use their internal degrees of freedom to reorient themselves. Classical examples include the falling cat system [1], [2] and satellite in orbit [3]–[6]. In this paper, we are interested in studying the motion of a highly articulated flying system: the Unified snake.

Inertial control of systems in the three-dimensional rotational space is difficult due to the global nonlinearity of this space, in addition to the noncommutivity of the structure of rotation group i.e. the order in which rotations occur cannot be interchanged. These systems are also usually underactuated which makes planning their motions difficult, especially for systems with high dimensional internal degrees of freedom.

Previous reorientation control for inertial systems in free flight has largely focused on either computational or geometric methods. Computational approaches, which directly integrate a system’s equations of motion, succeeded in generating controllers over a large range of motions. The drawback of these methods is that they are often black-box procedures which can potentially return poor solutions without incurring additional computational complexity. On the other hand, geometric approaches uses structure contained within the system to provide valuable insight into local control design. For systems in free flight, this structure is contained within a linear relationship, derived from conservation of angular momentum, that maps shape changes to resulting orientation changes through what is called “the reconstruction equation”[7]. Hatton

H. Salman is with the Department of Mechanical Engineering, American University of Beirut, Beirut, Lebanon e-mail: hcs03@mail.aub.edu

C. Gong and H. Choset are with the Robotics Institute at Carnegie Mellon University, Pittsburgh, PA, 15213, USA. emails: chaohuig@andrew.cmu.edu, choset@cs.cmu.edu

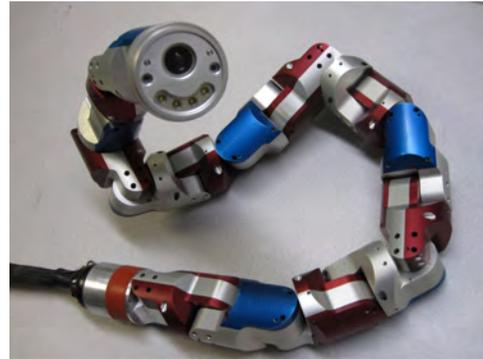


Fig. 1. An example of a Highly-Articulated System: the Unified Snake.

and Choset [8], [9] then addressed some local nonlinearity limitations, faced by previous geometric methods, by deriving a set of coordinates referred to as “minimum perturbation coordinates”. Recently, Travers et al. [7] brought the derivation of minimum perturbation coordinates to the space of three-dimensional rotations and were able to derive visual tools that provide the control designer intuition into selecting cyclic controllers for inertial systems in free flight.

Most of prior work analyzed underactuated systems with at most 2 internal degrees of freedom. Higher internal DOF’s systems are usually very difficult to control. In this paper, we introduce a new technique that uses what is called “Shape Basis” and benefits from prior geometric methods in order to control highly articulated systems i.e. systems with high dimensional shape space. We show how we can generate gaits that are capable of moving such a mechanical system in any desired direction. Finally, we apply these technique to a flying snake robot in order to generate gaits that reorient this snake robot in any direction

II. BACKGROUND

In this section we very briefly review some basic ideas and concepts from Lagrangian mechanics and mechanics of locomotion thoroughly covered in [10], [11].

A. The Reconstruction Equation

When analyzing highly-articulated locomoting systems, it is convenient to separate its configuration space Q into a fiber space G which represents the position of the system in space, and a base space M which represents the internal degrees of freedom of the system i.e. the shape of the system. This allows us to study how changing the shape of the system affects its

position. Thus, a configuration variable can be written as $q = (g, r) \in Q$ where g is the fiber variable and r is the base variable. The configuration velocity is then defined as $\dot{q} = (\dot{g}, \dot{r}) \in T_q Q$, the tangent space of the manifold Q at the configuration q .

For mechanical systems we can define the Lagrangian as a map from the tangent space of the configuration manifold to the reals. It is usually defined as the kinetic energy minus the potential energy of the system. But in this paper, we neglect the potential energy (since we are analyzing a floating snake system whose potential energy is zero) and thus we equate the Lagrangian of the system to its kinetic energy as follows:

$$L(q, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q} \quad (1)$$

where $M(q)$ is the mass matrix. Moreover, in this paper, we are dealing with systems that don't have any nonholonomic kinematic constraints i.e. the equation $w(q)\dot{q} = 0$ is trivially satisfied with $w(q)$, the matrix representing the kinematic constraints, being zero.

The independence of the location of the inertial frame when computing the Lagrangian leads to the invariance of the Lagrangian which allows us to compute the Lagrangian anywhere along the fiber space. The group identity was chosen in particular that is,

$$L((g^{-1}g, r), (T_g L_{g^{-1}} \dot{g}, \dot{r})) = L(q, \dot{q}) \quad (2)$$

where $T_g L_{g^{-1}}$ is a lifted action (a linear map acting on velocities) as defined in Shammass et al. [10]. Thus we can define the body representation of a fiber velocity \dot{g} to be $\xi = T_g L_{g^{-1}} \dot{g}$.

This allows us to compute the reduced Lagrangian, $l(\xi, r, \dot{r})$, which, according to Ostrowski (1995) [12], will have the following form

$$l(\xi, r, \dot{r}) = \frac{1}{2} (\xi \quad \dot{r}) \tilde{M} \begin{pmatrix} \xi \\ \dot{r} \end{pmatrix} \quad (3)$$

where \tilde{M} is the reduced mass matrix and will have the form

$$\begin{pmatrix} I(r) & I(r)A(r) \\ A^T(r)I^T(r) & m(r) \end{pmatrix} \quad (4)$$

where $A(r)$ is the local connection matrix, $I(r)$ is the local Inertia Tensor, and $m(r)$ is a matrix that depends only on the base variables. The importance of the reduced mass matrix is that it allows us to extract the local connection matrix (which will help us relate the body velocity ξ to the shape velocity \dot{r}) by manipulating sub-matrices of the reduced mass matrix.

The geometric mechanics community has addressed the question of relating the body velocity of the system ξ to the shape velocity \dot{r} . They developed what is called the *reconstruction equation* which is of the form

$$\xi = -\mathbf{A}(r)\dot{r} + \Gamma(r)p \quad (5)$$

where $\mathbf{A}(r)$ is as before the local connection matrix, $\Gamma(r)$ is the momentum distribution function, and p is the generalized nonholonomic momentum.

Shammass et al. [10] classified mechanical systems into three categories: (1) Purely Mechanical Systems, (2) Kinematic

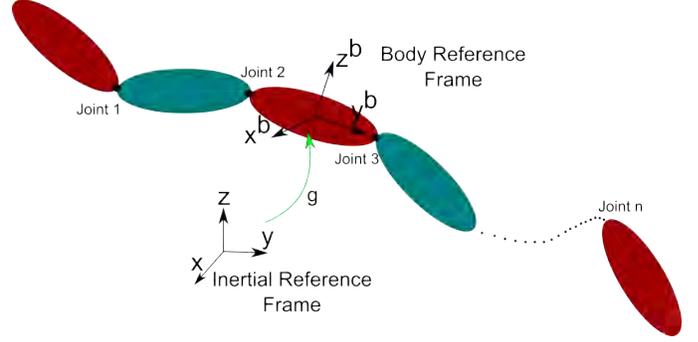


Fig. 2. A model of the floating snake that we are analyzing.

Systems and (3) Mixed Nonholonomic Systems while deriving the corresponding form of the reconstruction equation for each. In this paper, we are interested in analyzing a purely mechanical system whose reconstruction equation turns out to be of the form

$$\xi = -\mathbf{A}(r)\dot{r} \quad (6)$$

Thus, for purely mechanical systems, the body velocity of the system is totally determined by the shape velocity of the system. This encourages us to study how manipulating the shape of the system affects its position and how motion planning in the fiber space can be accomplished by planning in the base space.

B. Stokes' Theorem and Height Functions

Stokes' theorem equates the line integral along a closed curve on a vector on a space U to the integral of the of the vector field over a surface bounded by the curve,

$$\oint_{\delta\Omega} V(u) \cdot du = \iint_{\Omega} \text{curl} V(u) du \quad (7)$$

where Ω is a surface on U bounded by $\delta\Omega$ [9]

For the planer case where $U \in \mathbb{R}^2$, this theorem reduces to what is called Green's Theorem, and the curl of a vector field V on U is defined as:

$$\text{curl} V(u) = \frac{\partial v_2}{\partial u_1} - \frac{\partial v_1}{\partial u_2} \quad (8)$$

Based on this theorem, Shammass et al. [10] defines what is called a Height Function $F(u)$ on the shape space to be equal to $\text{curl} V(u)$. By plotting and analyzing the height functions associated with our system, we are able to identify gaits that produce desired displacements based on some rules of thumb that are presented in [10]

III. MATHEMATICAL MODEL

In this paper, we will be analyzing a highly articulated system, the Unified Snake floating in space. This system is basically made up of modules that are linked using alternating dorsal and lateral joints as shown in Fig. 1. We are interested in controlling the 3-Dimensional orientation of this snake in space since its center of mass will remain fixed under any change in the shape of the snake. This is due to the fact that no external forces act on the snake in the outer space.

In order to find the reconstruction equation and then start analyzing the effect of the shape of the snake on its orientation in space, we first model the snake and find its reduced mass matrix from which we can extract the local connection matrix as mentioned in Section II.

We will define the orientation of the snake as being the orientation of the body reference frame attached to one of the middle links as shown in Fig. 2. The configuration of this body frame with respect to an inertial frame is represented by the rigid body transformation $g \in SE(3)$. To represent this rigid transformation, we use 6 parameters: (x, y, z) representing the position of the body frame, and (α, β, γ) representing the three rotations along the three frame axes (pitch, roll, yaw) respectively. These variables constitutes the 6-dimensional fiber space of the system. The joint angles represent the internal degrees of freedom of the snake which are responsible for giving the snake its shape.

Hence, if the snake has n joints, the system has an $(n+6)$ -dimensional configuration space $Q = G \times M$ where the fiber space G is the special euclidean group $SE(3)$ representing the pose of the body reference frame attached to the snake. The base space $M = \mathbb{S} \times \mathbb{S} \times \dots \times \mathbb{S}$ (n -times) denotes the internal degrees of freedom of the snake.

The Lagrangian of this system, which is basically the Kinetic energy of the system, is invariant under group action as was discussed in Section II. So we choose to calculate the kinetic energy of the system using the body velocities of the links. We define the body velocity of the middle link, to which the body reference frame is attached, as $\hat{\xi}^0 := g^{-1}\dot{g}$ where $\hat{\cdot}$ is the wedge operator which takes a vector in \mathbb{R}^6 into the Special Euclidean space, $se(3)$, and $g \in SE(3)$ is as defined earlier and shown in Fig. 2. Also we define $\xi^i \in \mathbb{R}^6$, $i \in \{1, 2, \dots, n\}$, to be the body velocity of the other links.

Thus, the Lagrangian is defined as

$$L = \sum \xi^{iT} \mathcal{M}_i \xi^i, \quad (9)$$

$i \in \{0, 1, 2, \dots, n\}$, where \mathcal{M}_i is the inertia matrix of the i^{th} link. Using the rules for the transformation of velocities between reference frames, as defined in [13], we can express the body velocity of each of the links in terms of ξ^0 and the joint angles θ_j as follows:

$$\xi^i = Ad_{f_i^{-1}} \xi^0 + f_i^{-1} \dot{f}_i, \quad (10)$$

where $f_i \in SE(3)$ is the rigid body transformation between the i^{th} link and the body reference frame attached to the snake. Also note that $f_i^{-1} \dot{f}_i$ can be written as

$$f_i^{-1} \dot{f}_i = J \dot{\theta}, \quad (11)$$

where J is a $6 \times n$ jacobian matrix and $\dot{\theta} \in \mathbb{R}^n$ represents joint velocities.

Using (9), (10), and (11), we can derive the Lagrangian in the reduced form, $l(\xi^0, \theta, \dot{\theta})$, in terms of the fiber and the base variables. Then as discussed in section II, we can derive the local connection matrix, $\mathbf{A}(\theta)$, that relates the body velocity of the snake to its joint velocities according to the reconstruction equation in its simplified form:

$$\xi^0 = -\mathbf{A}(\theta) \dot{\theta} \quad (12)$$

As we mentioned earlier, we are interested in the controlling only the reorientation of the snake in space, so we will only take the last three equations of the system described in (12), which relate the body rotational velocities of the snake to the shape velocities, and we ignore the first three equations which relate the body translational velocities of the snake to its shape velocities. Hereafter, when we refer to (12), we consider a three-dimensional body velocity ξ^0 , related to the shape velocities $\dot{\theta}$ through a $3 \times n$ local connection matrix $\mathbf{A}(\theta)$.

IV. SHAPE BASED GAIT GENERATION

In order to control the reorientation of the snake through changing its shape, we define a gait G to be a closed curve in the shape space, M , of the snake. These gaits are set to be cyclic in order for the snake to retain its original shape after each period of time.

To see how shape changes reorient the snake, we integrate (12) with respect to time. For each row, the left hand side yields a change in the rotational body coordinates of the snake (body velocity integral), and the integrand of the right-hand side becomes a one-form [10]. Unfortunately, the one-form for the snake system does not simplify to the simple case of Green's theorem as discussed in section II. Thus we are not able to use the height functions techniques to generate gaits as shammass et al. [10] proposed. This is basically due to the high dimensionality of the shape space of the snake (n -dimensional). This paves the way for what is called "Shape Bases" in order to reduce the high dimensionality of the shape space and allow us to use the Height Functions techniques for motion planning.

A. Shape Basis

To reduce the high dimensional shape space of the snake, we choose to control the snake by giving it certain shapes i.e. we use a time variant function that depends on only two parameters to control the whole joint space. Thus by varying these two parameters, we control all the joints of the snake giving the snake a certain shape. The general form of such a function is:

$$r(n, t) = \theta(n, t) = \sigma_1(t) \beta_1(n) + \sigma_2(t) \beta_2(n), \quad (13)$$

where n is the joint index, t is the time, $r(n, t)$ is a base variables, σ_1 and σ_2 are two functions depending on time (together defines a parametrization of a gait in the base space), and $B = \{\beta_1, \beta_2\}$, which consists of two functions depending on the joint index n only, represents the "shape basis". For every gait, the shape basis remains the same, giving the U-snake some shape, whereas the two parameters σ_1 and σ_2 change with time in order to reorient the U-snake in a certain direction.

B. Example: Serpenoid Curve

As an example of a function that constitute of a shape basis, consider the serpenoid curve, which is widely used to control the U-snake and enable it to mimic the biological snakes' movements. One form of the serpenoid curve is given by:

$$\theta(n, t) = \sin(\omega t + \Omega n), \quad (14)$$

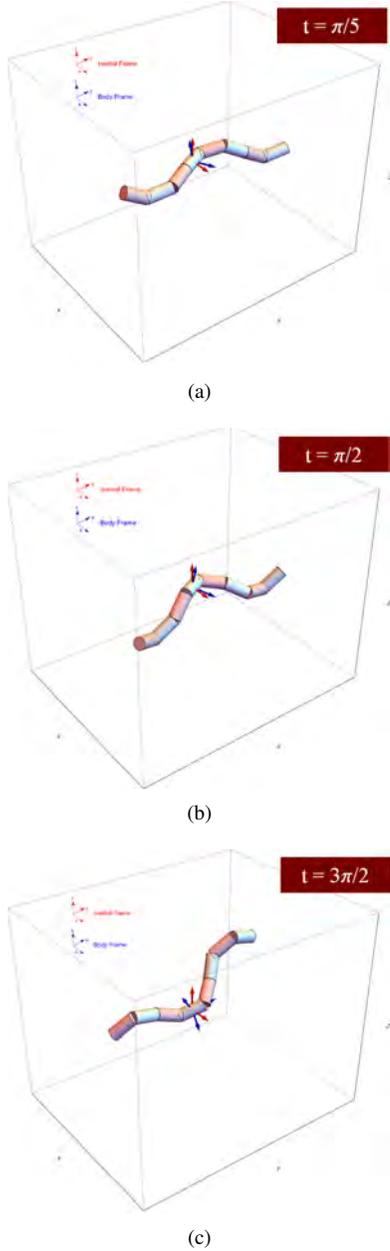


Fig. 3. The orientation of the snake at three different instants of times for a certain shape basis and a specific gait. Notice the same kind of shape that the snake takes at the three instants of time; that is how a shape basis works.

where ω is the temporal frequency, which depicts how much time the U-snake needs to repeat its shape, and Ω is the spacial frequency, which depicts after how many joints the U-snake repeats its shape. Using simple trigonometric relations, we can derive the following relation:

$$\theta(n, t) = \cos(\omega t) \sin(\Omega n) + \sin(\omega t) \cos(\Omega n). \quad (15)$$

Comparing (13) and (15), we see that the shape basis in this example is $\{\sin(\Omega n), \cos(\Omega n)\}$, and the controlling parameters σ_1 and σ_2 are $\cos(\omega t)$ and $\sin(\omega t)$ respectively.

To have an idea of what a shape basis yield to, consider a U-snake consisting of 8 links. We control the snake by using the shape basis $B = \{\sin(\Omega n), \cos(\Omega n)\}$ with $\Omega = \frac{\pi}{3}$, and consider the gait defined by the parametrization $\{\sigma_1 =$

$\frac{\pi}{8}(2 \sin(t) + \sin(2t)), \sigma_2 = \frac{\pi}{8}(2 \sin(t) - \sin(2t))\}$. Three different snapshots at three different time instants are taken for this simulation and are shown in Fig. 3. Notice that the snake takes a certain shape during the simulation which is directly a result of the shape basis choice.

C. Height Functions

The most important advantage of a shape basis is that it enables us to regain the ease of using height functions techniques to generate gaits that drives the system in a desired direction. This is due to the fact that the shape basis technique maps the n -dimensional base space to a two-dimensional base space. By using the chain rule, we can reformulate the integral of the reconstruction equation as follows:

$$\int_{t_0}^{t_1} \xi^0 dt = - \oint_G \left[\mathbf{A}(\theta) \frac{\partial \theta}{\partial \sigma} \right]^{3 \times 2} d\sigma, \quad (16)$$

where theta is as defined in (13) and $\sigma = (\sigma_1, \sigma_2) \in \mathbb{R}^2$ is the new base controlled shape space. Notice that the quantity inside the brackets is a 3×2 matrix. Thus, the use of a shape basis converts the hard problem of solving the integral in (16) using Stokes' Theorem to an easy problem that is solved using Green's theorem as discussed in Section II. The Height functions in this case are defined as the curl of each of the rows of the matrix inside the brackets. Thus each shape basis leads to three height functions. After that, we analyze these height functions and generate gaits that will lead to desired motions similar to what Shammam et al. [10] did.

V. GAITS FOR REORIENTING THE FLOATING SNAKE

As an application of the above techniques, we choose to simulate an eight link U-snake floating in space. We are interested in rotating the snake in the pitch, roll, and yaw directions, so we generate three different gaits (G_1, G_2, G_3) that fulfill these motions.

We choose to use the shape basis that is used in a serpenoid curve $\{\sin(\Omega n), \cos(\Omega n)\}$, but we vary the spacial frequency until we get nice height functions that allow us to build the appropriate gait for the appropriate motion. For example, in order to rotate the snake in the pitch direction, we choose the spatial frequency $\Omega = \frac{\pi}{3}$, and we generate the gait G_1 represented by a parametrization $\{\sigma_1 = \frac{\pi}{8}(2 \sin(t) + \sin(2t)), \sigma_2 = \frac{\pi}{8}(2 \sin(t) - \sin(2t))\}$ in a way so that it envelops a nonzero volume only under the first height function corresponding to the first row of the 3×2 in (16). The three height functions, corresponding to this case, are shown in Fig. 4 as the first three rows of the first column. The green color reflects positive regions whereas the blue color reflects negative regions.

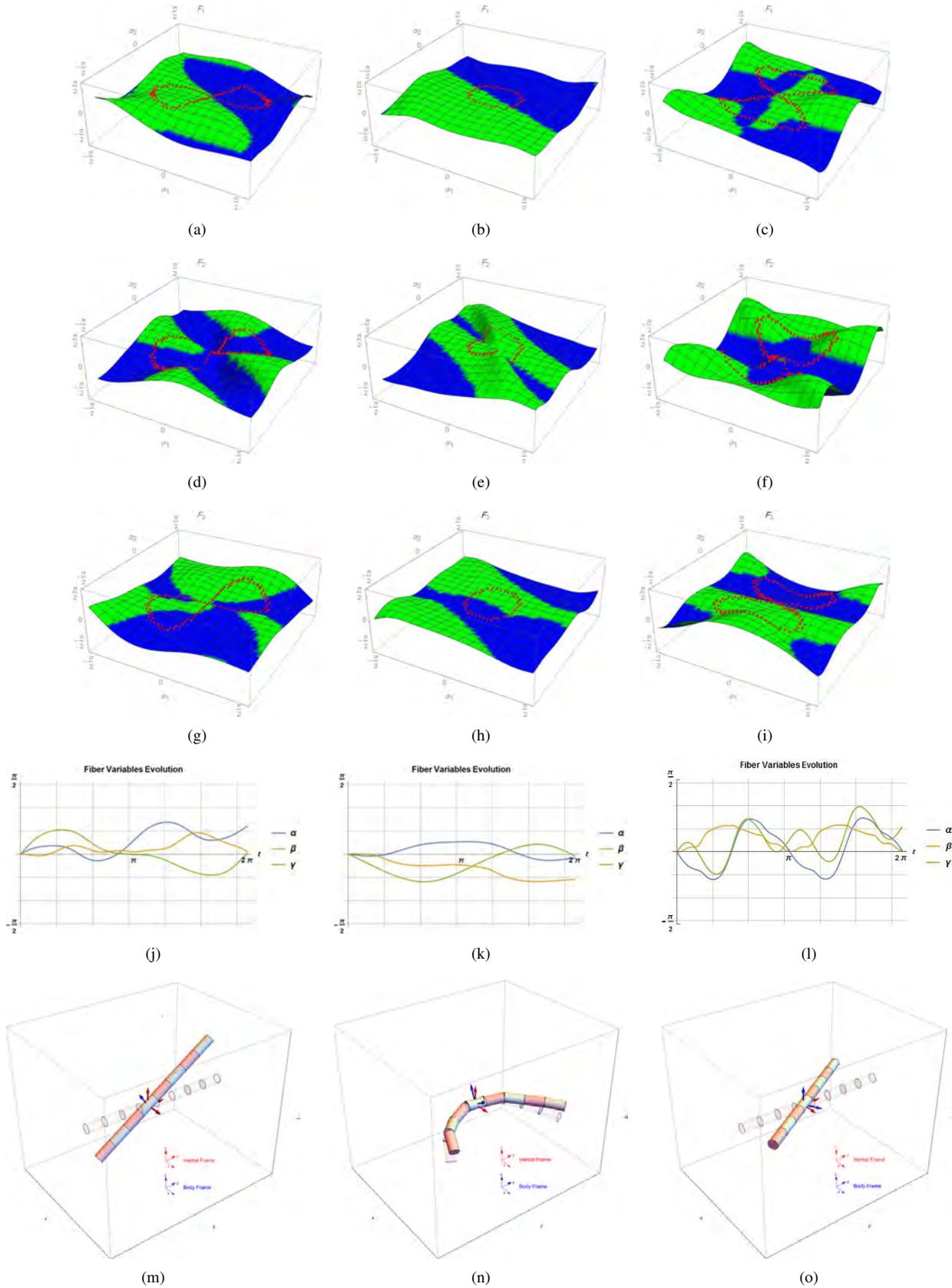


Fig. 4. Three sample gaits, (G_1, G_2, G_3) that were designed to rotate the U-snake along each of the three rotation axes. The first column depicts a gait that rotates the U-snake around the x-axis (pitch), while the second and third rows depict gaits that rotate the U-snake around the y-roll and z-(yaw) axes, respectively. The first three rows of each column depicts the three height functions associated with the snake for each of the gaits. The fourth row of each column depicts a time simulation of each gait where the rotations around each axis are plotted versus time. The last row of each column shows the initial and final orientations of the U-snake for each gait represented by the transparent and opaque graphics respectively.

The forth row represents the evolution of the fiber variables as a function of time. Notice the nonzero change in α (pitch) and the zeros net change in β (roll) and γ (yaw) at the end of the simulation. Similarly we generate gaits to rotate the snake in the roll an yaw direction. The corresponding characteristics if the three gaits listed in Table 1.

TABLE I
TABLE OF INPUTS USED FOR EACH GAIT. THE SHAPE BASIS USED IS $B=\{\sin(\Omega n), \cos(\Omega n)\}$

Gait	Ω	$\sigma_1(t)$	$\sigma_2(t)$
G_1	$\pi/3$	$\frac{\pi}{8}(2\sin(t) + \sin(2t))$	$\frac{\pi}{8}(2\sin(t) - \sin(2t))$
G_2	$\pi/16$	$0.7(\sin(2t))$	$-0.7(\cos(2t))$
G_3	$\pi/2.6$	$0.85\sin(4t)$	$\sin(2t)$

Therefore, we are able to reorient the snake in any desired direction as depicted in Fig. 4.

VI. CONCLUSION

In this paper, we presented a new way to deal with the high dimensionality of the shape space of highly articulated systems. We defined what is called a Shape Basis and showed how this basis can reduce the base space from its high dimension to a two-dimension base space for which it becomes easy to apply the Height functions techniques for motion planning proposed by Shamma et al. [10]. We applied this new technique on a Unified snake floating in space in order to reorient it in any direction we want. Finally we built three gaits that rotates this snake in the pitch, roll and yaw directions simply by looking at the height functions associated with the system.

REFERENCES

[1] R. Montgomery, "Isoholonomic problems and some applications," *Comm. Math. Phys.*, vol. 128, pp. 565–592, 1990.

[2] —, "Gauge theory of the falling cat, ser," *Dynamics and Control of Mechanical Systems. American Mathematical Society*, 1993.

[3] G. Walsh and S. Sastry, "On reorienting linked rigid bodies using internal motions," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 1, pp. 139–146, 1995.

[4] Z. li and L. Gurvits, "Theory and application of non-holonomic motion planning," *Courant Inst. Math and Science, New York University, Tech. Rep.*, 1990.

[5] C. Byrnes and A. Isidori, "On attitude stabilization of rigid spacecraft," *Automatica*, vol. 27, pp. 87–95, 1991.

[6] Y. Nakamura and R. Mukherjee, "Exploiting nonholonomic redundancy of free-flying space robots," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 4, pp. 499–506, 1993.

[7] M. Travers, R. Hatton, and H. Choset, "Minimum perturbation coordinates on $so(3)$," *Proc. 2013 American Control Conference, Washington, DC*, 2013.

[8] R. Hatton and H. Choset, "Optimizing coordinate choice for locomoting systems," *In Proceedings of the IEEE International Conference on Robotics and Automation, Anchorage, AK*, 2010.

[9] —, "Geometric motion planning: The local connection, stokes' theorem, and the importance of coordinate choice," *International Journal of Robotics Research*, vol. 30, no. 8, pp. 998–1014, 2011.

[10] E. Shamma, H. Choset, and A. A. Rizzi, "Geometric motion planning analysis for two classes of underactuated mechanical systems," *International Journal of Robotics Research*, vol. 26, no. 10, pp. 1043–1073, 2007.

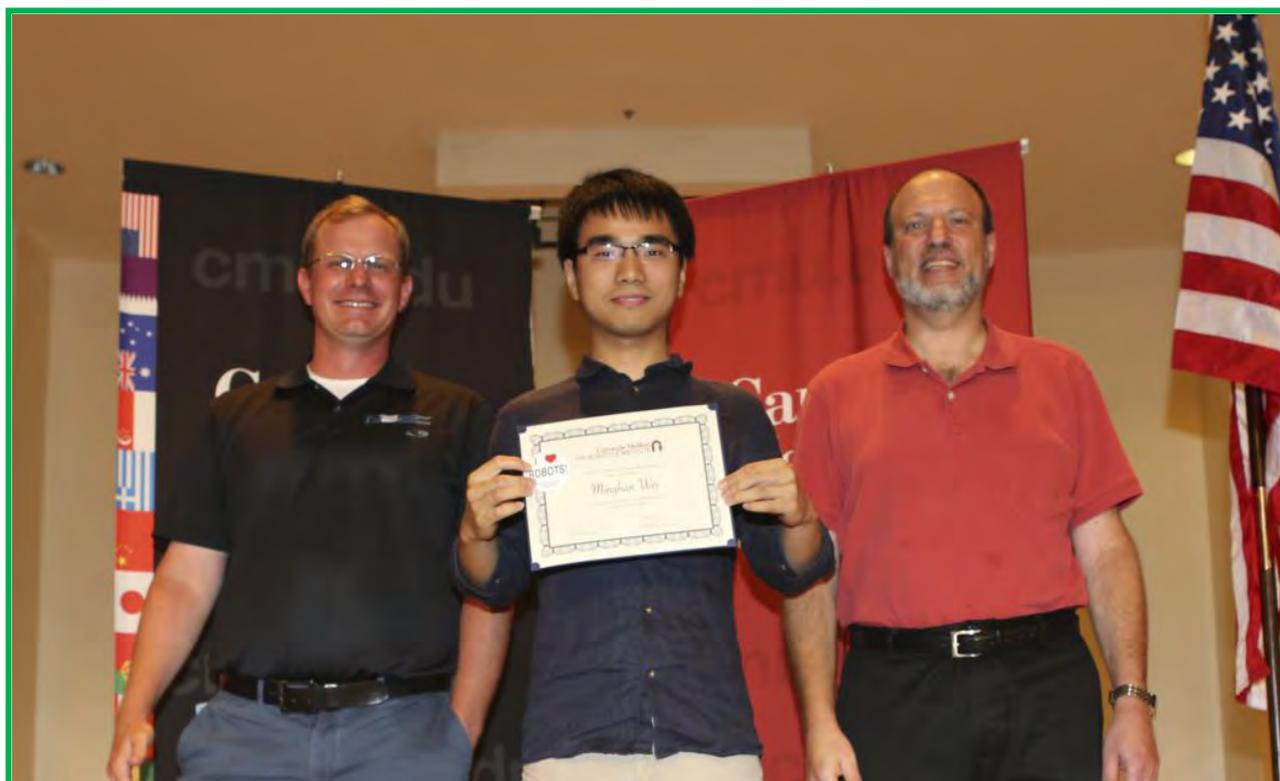
[11] B. AM, *Nonholonomic Mechanics and Control*. Berlin: Springer, 2003.

[12] J. Ostrowski, J. Burdick, A. Lewis, and R. Murray, "The mechanics of undulatory locomotion: The mixed kinematic and dynamic case," *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, vol. 2, pp. 1945–1951, 1995.

[13] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

Minghan Wei

RISS 2015



Detecting Vehicles Based On Smartphone-collected Images

Minghan Wei

Abstract: the Navlab group has developed a smartphone-based system for infrastructure inventory and assessment. The system collects images from a vehicle and analyzes the images to find road distress and other infrastructure problems. One issue with the collected images is that they often contain vehicles. These vehicles not only obstruct the view, but there are also privacy concerns. It is desirable to either discard images with vehicles or to black out the corresponding areas in the images. In this work we implement and train a vehicle detector using standard computer vision tools to solve that issue. The detector is reasonably fast and performs comparable to the state-of-the-art. We also determined how many training examples we need to use to train a classifier with good performance.

Index Terms—vehicle detection, Haar, Adaboost, smartphone, OpenCV

I. INTRODUCTION

IT is a tedious and error-prone work to assess road damage. This is currently done by human inspectors. By developing a machine vision algorithm based on smartphone-collected video to replace this work for human inspectors, we can monitor city road conditions automatically with a much lower cost. An algorithm using that method has been developed by Navlab [1]. However, vehicles on road tend to obstruct the view as well as raise privacy concerns. The location of the vehicle at a given time is personal information. When we build our road dataset we do not want to include that information in it. To address this issue, we can use another machine vision algorithm to detect vehicles and then black out the regions covered by vehicles or simply delete those images with vehicles. In this way we can not only improve the accuracy, but also the privacy of other drivers is respected.

Vehicle detection by itself is an important issue of computer vision. Besides vision, other ways such as radar and lidar [2] for vehicle detection have also been studied in literature. Though radar sensing works well for detecting and tracking preceding vehicles in the ego lane, it cannot provide wide field-of-view. As we want to detect all the vehicles in the view, radar is not suitable for our application. Lidar sensing provides a much wider field-of-view than radar and is not sensitive to noise

Minghan Wei is with College of Elite Education of Nanjing University of Science and Technology in Jiangsu Province, China. His major is Computer Science and Technology. Email: minghanwei19@gmail.com.

This research was conducted at Carnegie Mellon University under the guidance of Christmas Mertz.

except precipitation. The influence of precipitation can be solved by collecting data at different time and weather, but the cost remains a problem. Though the price of lidar sensing continues to drop these years, it will still be too expensive to install lidar sensing on all vehicles described in [1] to collect road information data. So the choice of lidar is rejected as well. As the Navlab project uses a machine vision algorithm to detect road distress automatically, we will focus on the detectors based on vision.

There have been several well-designed algorithms to detect vehicles. The performance of ‘HOG+SVM’ and ‘Haar + Adaboost’ in detecting vehicles is examined in [3]. [4] shows combining integral channel features with a standard boosting algorithm to detect pedestrians can outperform other existing algorithms, from which we can expect ICF could also have better performance when dealing with vehicles. [5] described an object detection system based on mixtures of multiscale deformable part models. It gets good performance on PASCAL 2006 database.

As mentioned, our primary goal is to black out regions in an image covered by vehicles or to discard images with vehicles in them as a pre-processing step for road distress analysis. In this application, high recall rate is important as we want to black out all the regions of vehicle or to discard all images with vehicles. We get road data from the same stretch of road many times and these images have overlapped areas. We can disregard many images so that higher false positive are acceptable. The speed of processing images should be less than one second on each image. We use OpenCV, a free open source library, as our computer vision tools to implement the algorithm. Precision-Recall curves are drawn to evaluate the performance. We compare it with the performance of [3]. In the evaluation we also note how many training samples we used, as an indication of how many samples are needed to come out with a practical detector.

II. FEATURE AND LEARNING ALGORITHM

The method to detect objects is to extract Harr features of a certain class and then use Adaboost learning algorithm.

A. Haar Wavelet

Research in computer vision has shown that Haar feature is powerful for object recognition [6]. We do a brief review of

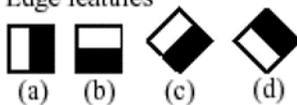
Haar feature here.

A Haar-like feature can be defined as the difference of the sum of pixels of areas, which can be at any position and scale within the original image. The relative position of areas is shown as Figure 1. The values indicate certain characteristics of a particular area of the object of a certain class. Then these values can be used to represent an object class and learned by a machine learning algorithm.

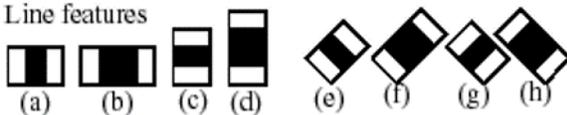
When the edges of the rectangles used to calculate Haar features are parallel to the edges of the boundary of the image, (such as 1(a), (b) in Figure 1), Haar features can be quickly calculated using integral image [7]. Let I be the original image. Then the integral image, II , at x, y contains the sum of pixels above and to the left of x, y .

$$H(x, y) = \sum_{x' < x, y' < y} I(x', y'). \quad (1)$$

1. Edge features



2. Line features



3. Center-surround features

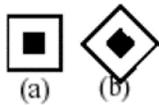


Figure 1: the relative position of two areas which are used to calculate Haar feature.

With integral image, we can calculate the sum of a region rapidly. Take Figure 2 as an example:

$$sum = II(c) - II(B) - II(D) + II(A) \quad (2)$$

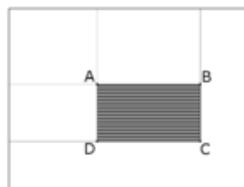


Figure 2: An example of rectangles used to calculate Haar features.

B. Adaboost

Adaboost algorithm trains a cascade classifier consisting of subsequent weak classifier. Each weak classifier can be seen as a stage. A window will be rejected immediately when it fails in one stage, which will speed up the detection because not all windows need to be calculated in all stages. Only windows which pass all the stages will be regarded as a target. More details for Adaboost can be learned in [9].

III. PREPARING TRAINING SAMPLES

To use OpenCV function to train a classifier, we need to prepare samples and make description files in a certain format. A detailed instructions for how to prepare can be found online at [10]. Here we provide some additional details for our case.

Figure 3 shows four positive samples we collected. Our positive samples are mainly from the KITTY database which is available at [8]. We crop out vehicles using the ground truth in the database. We do this because not all vehicles in the dataset is suitable as training samples. Some vehicles in Kitty database are too small. Some others look too dark due to the shadow. After we have cropped all vehicles and chosen those suitable as positive samples, it is easy to generate the text file which describe the information of these vehicles using C code.



Figure 3: four positive samples. We collect samples from front, back and side views from a vehicle.

All the positive samples will be resized to a user-specified size before extracting Haar features. Complete Haar features can be saved if we extract Haar features at the original size and in that case the features we get are the most representative. However, the size of Haar features to be extracted from the image will increase exponentially with the increase of the size of the image. If we choose original size or the average of the original size of all positive samples, the training process will take too long. On the other hand, different vehicles tend to have different appearance. If we compute Harr features at a lower resolution, it is easier to get some common features of different vehicles. Our experiment shows that 24*18 pixels (width* height) is a good choice

IV. TRAINING AND DETECTING

Detailed instructions of how to use this function can also be found at check [10]. The number of negative and positive samples used in each stage should be a little less than the total number we collected. In the evaluation part, we will discuss how to optimize these parameters used in the training function to generate a better classifier.

The following diagram shows the flow chart of the detection work. We implemented it in C code.



Figure 4: The flow chart of the program

When detecting vehicles, a sliding window will move across the image of different scales. We amended the original detection codes of OpenCV so that the detection result could be more accurate. A sample result is shown in Figure 4.

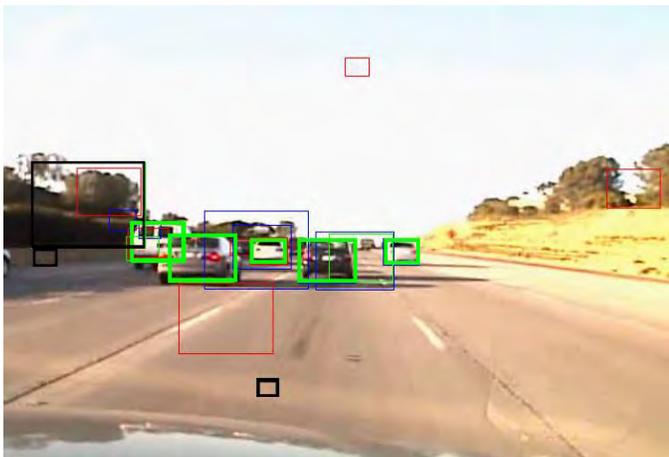


Figure 4: Green, thick bounding boxes indicate a detection with high confidence. Other boxes' confidences are lower. A precision-recall curve could be draw by adjusting the threshold for detection.

V. EVALUATION OF THE PERFORMANCE

We trained a few classifiers with different parameters. We do this because we want to examine by experiment how to get better classifiers with OpenCV. Our test set is from a publicly available dataset which can be found online at: [10]. This is a relatively challenging dataset. It is captured in rush hour. There are many vehicles to be detected and the shadow may cause false detection. It takes 0.6s to process a 704*480 image. Figure 6 shows the performance of the classifiers we trained. We tested on this database to see if our algorithm could get good results when applying to a different environment.

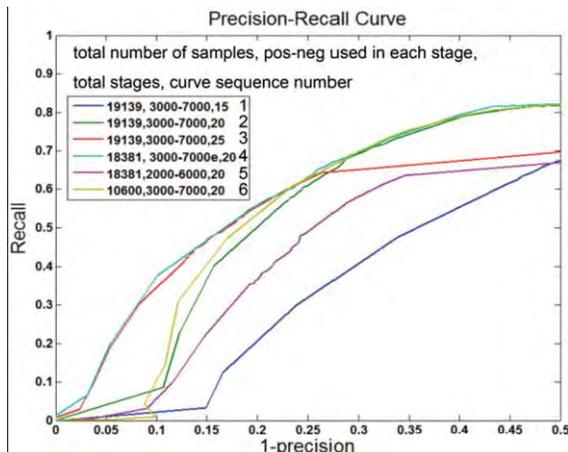


Figure 6: the performance of the classifiers we trained. We will explain why we choose '1-precision' as x-axis later.

We use PASCAL criteria [12] to evaluate our results. That is: a detection is regarded as correct if:

$$\frac{Area_{ground_truth} \cap Area_{detection_result}}{Area_{ground_truth} \cup Area_{detection_result}} > 0.5 \quad (3)$$

The following table shows the differences between these classifiers.

Sequence number	Positive samples used in each stage	Negative samples used in each stage	Number of stages	Total samples (positive+ negative)
1	3000	7000	15	19139
2	3000	7000	20	19139
3	3000	7000	25	19139
4	3000	7000	20	18381
5	2000	6000	20	18381
6	3000	7000	20	10600

Table 1: the differences between the classifiers in Figure 6

From these curves we can conclude:

- Number of samples used in each stage will make a big difference. More samples used in each stage, better the performance will be (compare curve 4 and 5).
- More training stages does not guarantee better results (compare curve 1, 2, 3). In our application, 20 stages is preferred due to its higher recall.
- On the one hand, compared with number of samples used in each stage, the total number of samples plays a less significant role (compare curve 2, 4). On the other, our positive samples should be representative (not too dark, even sunlight) for vehicles. We delete some positive samples which are labeled as vehicle in the original database but either too dark or truncated from the training set from a total of 19139 to 18381. Overall the training set of 18381 have better performance.

To make a comparison between our curves and the start-of-the-art Haar + Adaboost classifier and examine if OpenCV can make the best of Haar + Adaboost, we put together our best curve with curves from the literature [3]. Figure 7 shows the comparison. Because in [3] the x-axis is '1-precision' and we want to put the original curve from [3] into our figure, we also use '1-precision' as our axis here.

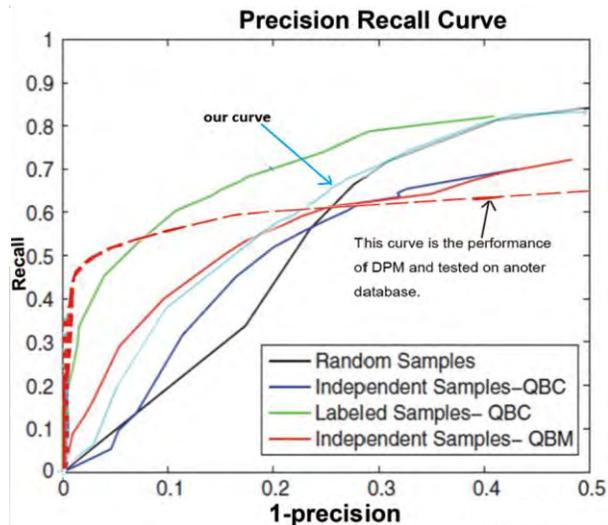


Figure 7: The best curve we with the performance from a literature. Another curve from [5] using DPM and tested on another database is also provided.

The ‘QBC’ and ‘QBM’ are different strategies to query samples for retraining. In [3], an initial classifier is trained and then adds more samples to the training set to retrain the classifier. One way to get more samples is called ‘query by confidence (QBC)’. Windows with a confidence near the deciding boundary are usually the most informative. The initial classifier was tested on a database and those windows with a confidence close to deciding boundary was add to the training set. The other strategy, ‘QBM’, which stands for ‘query by misclassification’, also tests the initial classifier on a new database but needs human labor to pick out those windows which are classified wrongly. Those false detections will then be added to the training set. The process of querying samples using ‘QBC’ takes less time than ‘QBM’, cause ‘QBM’ usually needs human monitors to decide if a window is classified wrongly.

Our classifier in Figure 7 is an initial one which does not take advantage of retraining. The performance of our classifier is better than the black and purple curve in Figure 7, comparable to the black one. Though it is not as good as the green one, we can expect that the classifier we trained using OpenCV function could gain performance as good as state-of-the-art Haar + Adaboost classifier by retraining. We also include the performance of a classifier trained with deformable parts model algorithm [5]. It is tested on another database.

Figure 8 and 9 shows the typical failure cases of our classifier. We discuss these failure cases here to see if these false detection will significantly undermine our application.



Figure 8: The example of false positive



Figure 9: The example of false negative

False detection are usually caused by vegetation along the road. The area interested in [1] is road. So the false positive outside the road area will not affect the analysis. Meanwhile, we found that in a video the false positive in a frame will not appear in the next frame. By relating the detection results of adjacent frames this kind of false could be solved. The other kind of fault is false negative. For example, in Figure 9 two vehicles are missed. Most vehicles are missed because they are far away from the view and look too small. By limiting road distress analysis to areas close to the view, those missed vehicles will not matter much. Since the vehicle which is described in [1] to collect data will drive along the road, limiting the analysis area will not miss any place on the road.

VI. CONCLUSION & FUTURE WORK

By experiment we found that OpenCV is an effective tool to do detection work. The feature OpenCV used is Haar and it is learned using Adaboost learning algorithm. The performance is comparable to state-of-the-art Haar+Adaboost classifier and satisfies our application.

We still want to get better performance. One way we can do this is to take advantage of the retraining process. Using querying by confidence or querying by misclassification, we are able to collect more training samples to train better classifiers.

Another way we can do this is to seek other methods to detect. DPM [5] has the potential to outperform Haar + Adaboost classifier. It is not implemented in OpenCV. But another computer vision library which can be found online [11] has implemented them. We will try to prove that.

ACKNOWLEDGMENT

I would like to sincerely thank Christoph Mertz for his kind help during this project.

REFERENCES

- [1] Christoph Mertz, Srivatsan Varadarajan, Sobhagya Jose, Karan Sharma, Lars Wander and Jina Wang., “City-wide Road Distress Monitoring with Smartphones”, Proceedings of ITS World Congress, September, 2014.
- [2] Sayanan Sivaraman, Mohan Manubhai Trivedi, “Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis”, *IEEE Transaction on Intelligent Transportation Systems*, VOL. 14, NO. 4, December 2013.
- [3] S. Sivaraman and M. M. Trivedi, Active learning for on-road vehicle detection: A comparative study, *Mach. Vis. Appl*, vol. 16, pp.1 -13 2011.
- [4] P. Dollar, Z. Tu, P. Perona, and S. Belongie, ‘Integral Channel Features”, *BMVC* 2009.

- [5] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. "Object Detection with Discriminatively Trained Part based Models". *IEEE Trans. on Pattern Analysis and Machine Intelligence*.
- [6] J. Whitehill and C.W. Omlin, "Haar Features for FACS AU Recognition," *Proc. IEEE Intl Conf. Automatic Face and Gesture Recognition (AFGR '06)*, pp. 217-222, 2006.
- [7] P.A. Viola, M.J. Jones, "Rapid object detection using a boosted cascade of simple features", in: *CVPR*, issue 1, 2001, pp. 511-518.
- [8] http://www.cvlibs.net/datasets/kitti/raw_data.php.
- [9] http://docs.opencv.org/doc/user_guide/ug_traincascade.html.
- [10] <http://cvrr.ucsd.edu/LISA/index.html>.
- [11] <http://libccv.org/>
- [12] J.Ponce, T.L. Berg, M Everingham, D. Forsyth, M. Hebert, Dataset issues in object rec. In *Towards Category-Level Object Rec.*, pages 29-48. Springer, 2006.

Xingchen Yu

RISS 2015



Methods and Experiments of Pedestrian Detection

Xingchen Yu
Robotics Institute Summer Scholar
Carnegie Mellon University
Pittsburgh, U.S.
xingchen9382@gmail.com

Abstract—This paper addresses the development and methods of pedestrian detection we used, and couple of experiments based on these methodologies. Pedestrian detection is a key problem in computer vision, with several applications that have the potential to positively impact quality of life and pedestrian detection was developed rapidly during last decades. HAAR feature and HOG feature are two important features for object detection. INRIA and CVC are two of outstanding pedestrian databases which used in many famous experiments of pedestrian detection. We utilize these two features and these databases to train our pedestrian classifiers and analyze images under OpenCV. We evaluate results using miss rate and false positive per image. Our study provides a framework for gauging future efforts. Our experiments show that performance still has much room for improvement. In particular, detection is disappointing at low resolutions and for partially occluded pedestrians.

Keywords—pedestrian detection; computer vision; HAAR; HOG; classifier; OpenCV; miss rate; false positive per image;

I. INTRODUCTION

Pedestrian detection is defined as: check pedestrians in the input images (or video frames), if there is, give location information. It is the first step of vehicle auxiliary driving, intelligent video surveillance and human behavior analysis. In recent years, pedestrian detection is also used in the emerging fields, such as aerial images, victim rescue. Pedestrians have both rigid and flexible object characteristics, and the appearance is easy to be influenced by wearing, scale, shielding, attitude and angle of view, which makes it difficult and hot spot to develop pedestrian detection^[1].

The topic of this RISS project is pedestrian detection in images. There are many machine vision approaches for detecting pedestrians. We want to use OpenCV tools to implement, train, and evaluate detectors using Haar and Hog features. Under the environment of opencv, we need to collect positive and negative images, train cascade classifier, detect

and localize upright people in static images.

Challenges of pedestrian detection:

- a. Wide variety of articulated poses
- b. Variable appearance/clothing
- c. Complex backgrounds
- d. Unconstrained illumination
- e. Occlusions, different scales

Applications of pedestrian detection:

- a. Pedestrian detection for smart cars
- b. Film & media analysis
- c. Visual surveillance

II. DATASETS

We trained our classifier with three different positive datasets. And we collected large number of negative samples from Internet.

A. INRIA pedestrian database

The first is the well-established INRIA pedestrian database, containing 3506 64×128 images of humans cropped from a varied set of personal photos. It contains various views with a relatively large range of poses. Fig. 1 shows some samples. The people are usually standing, but appear in any orientation and against a wide variety of background image including crowds. Many are bystanders taken from the image backgrounds, so there is no particular bias on their pose.



Fig. 1. Positive samples, which are 64×128 images of humans cropped from a varied set of personal photos.

B. CVC02 pedestrian database

CVC-02 consists of 2054 images of humans. The imagery has been recorded in urban scenarios around Barcelona (Spain), using a Bumblebee color stereo camera with resolution 640×480 pixels and 6mm focal length. The annotated pedestrians are in the range from 0 to 50 m from the camera, which corresponds to a smallest pedestrian of 12×24 pixels.

C. CVC07 pedestrian database

This dataset contains 4824 pedestrian images. The pedestrian images have frontal view and left view, which are annotated as ‘M’ and ‘L’. You may flip the pedestrians to get right view examples. Part annotations are also provided.

III. METHODS

A. Training the classifier

At first, we need to use enough positive and negative images we collected before. We put them into separate folders. For convenience, we should use numbers to name these images in order properly. Secondly, for each folder, we need to create a txt file to describe images in it. Each line of the contents in ‘pos.txt’ (the txt to describe positive images) should be like this: whole image’s name, number of targets x position(left top) and y position(left top), width, height. The number of lines of the file should correspond to the number of images. The txt file to describe negative images is simpler. We only need to include name of the images in each line. Then we need to use openCV function ‘opencv_createsamples’ utility to create positive samples. At last, we train the cascade classifier. The function to train classifier is called ‘opencv_traincascade’. The basic parameters are explained here:

- data: the path to store the output classifier.
- vec: the path off the positive description file we get from last step.
- bg: the path of the txt describing negative samples.
- numStages: number of cascade stages to be trained.
- w -h : should be the same as last step.
- numPos: number of positive samples in each stage. It should be smaller than the whole quantity of positive samples.

-numNeg: similar to numPos.

numPos, numNeg, numStages will affect the time of training and performance of the classifier.

Usually, the larger of these parameters, the longer time consumed to train, and the better performance we get.

Now we can get a classifier named ‘cascade.xml’.

B. Haar-like feature

Haar-like Feature are the difference of the sum of pixels of areas inside the rectangle, which can be at any position and scale within the original image.

Haar-like features are divided into three categories: edge features, linear features, central features and diagonal features, which are composed of feature templates. Fig. 2 shows these three categories. The feature template has two rectangles: white and black. Haar eigenvalues reflect the gray level of the image. Such as facial features can be described by a simple rectangle features, such as: The color of eyes is darker than the color of the cheek, the color of both sides of the nose is darker than the color of nose, the mouth is darker than the surrounding color. But the rectangular feature is sensitive to some simple graphics, such as edges and line segments, so it can only describe the structure of a specific trend (horizontal, vertical, diagonal).

One of the commonly used techniques is the integral image. Integral image is a matrix of the same size as the original image, and the value of each element is the sum of all the pixels in the upper left corner of the image.

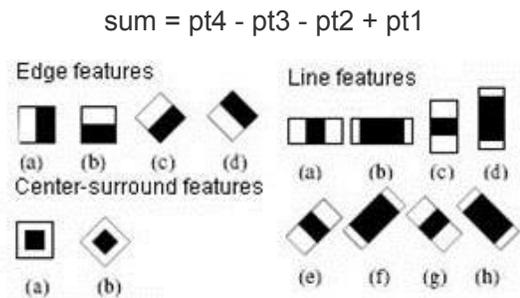


Fig. 2. Three categories of Haar-like feature.

C. Hog feature

The full name of Hog is histogram of oriented gradient which is a feature descriptor used for target detection. This technique counts the number of times of the image local orientation gradient. The method is similar to the edge direction histogram and scale-invariant feature transform, but the difference is that the hog is based on the consistent space of the density matrix to improve the accuracy. Navneet Dalal and Bill Triggs proposed HOG in 2005 CVPR for the first time, in order to use it in static image or video of pedestrian detection^[2].

The core idea of HOG is that the shape of the detected object can be described by the intensity gradient or the edge direction.

The whole image is divided into small connected regions (cells), each cell generating an orientation gradient histogram or the edge direction of pixel in cell. The combination of these histograms can express descriptors. In order to improve the accuracy, the local histogram can be standardized by calculating the intensity of a large area (block) in the image as measure. And use this value (measure) to normalize all cells in this block. Fig. 3 shows the processing chain of hog-svm classifier training and Fig. 4 shows the processing of hog feature extracting.

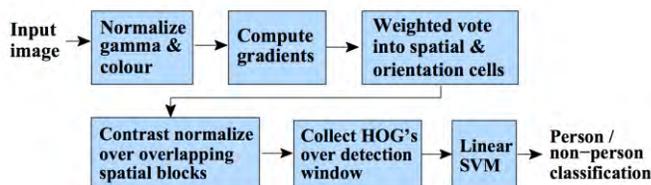


Fig. 3. Processing chain of hog-svm classifier training.

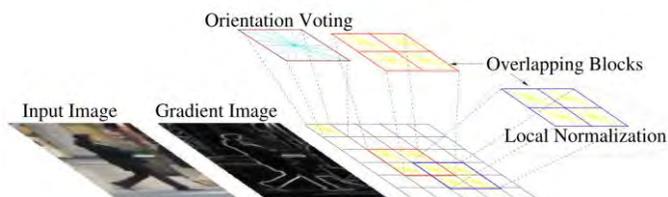


Fig. 4. Processing of hog feature extracting.

IV. EXPERIMENTS

A. Overview of experiments

Based on Haar-like feature, we tested the dependency of the classifiers on the number of training samples and the number of stages. We did four experiments and got their results. And we used miss rate and false positive per image to express the detection results. We also compared our results with the results in the paper^[1]. In the paper, they evaluated different methods to detect pedestrians. The average analysis time per image is 300ms.

In the first and second test, we used the same database consisted of INRIA and CVC02, and the number of stage in these two tests both are 20. We used 1000 positive images and 8089 negative images in the first test and we took advantage of 800 positive images and 2400 negative images per stage. In the second test, we used 5560 positive images and 16552 negative images, and the numbers of positive and negative images per stage are 4000 and 12000. In the third and fourth test, we chose a different database. It is made up of INRIA and CVC07. We used 8330 positive images, 29552 negative images, 7000 positive and 21000 negative images per stage in third and fourth test. The number of stage in the third test is 20, but in the fourth test, we used 25 stages.

TABLE I
DATA OF TESTS

Test	Database	Total Pos Num	Pos Num per Stage	Total Neg Num	Neg Num per Stage
1	INRIA&CVC02	1000	800	8089	2400
2	INRIA&CVC02	5560	4000	16552	12000
3	INRIA&CVC07	8330	7000	29552	21000
4	INRIA&CVC07	8330	7000	29552	21000

Test	Training Time	Stage	Curve Color
1	12 hours	20	red
2	42 hours	20	green
3	73 hours	20	blue
4	95 hours	25	orange

Additionally, we used an OpenCV pre-trained classifier called 'hogcascade_pedestrians.xml' to test the hog feature. The results of all experiments are showed in the next part.

We tried to extract hog feature from images and trained an hog-svm classifier. However we did not get a satisfactory result for some reasons, such as time, algorithm, etc.

B. Results of experiments

Fig. 5-Fig. 9 shows the examples of result of our five tests. The green rectangles express the detected pedestrians. Pedestrians without green rectangles are false positive which we lost in the detections.

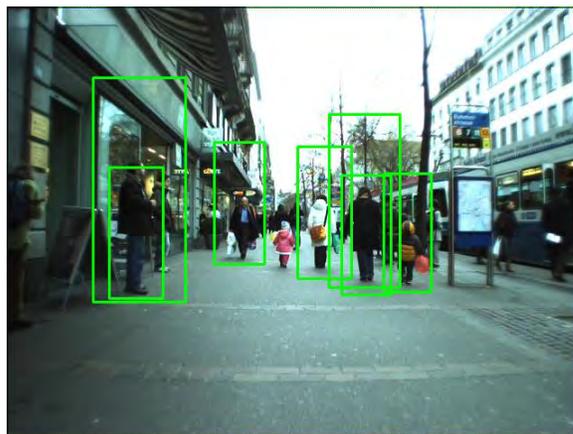


Fig. 5. First test example.

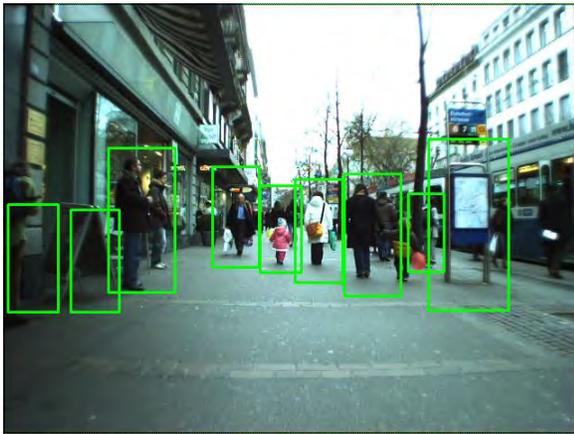


Fig. 6. Second test example.

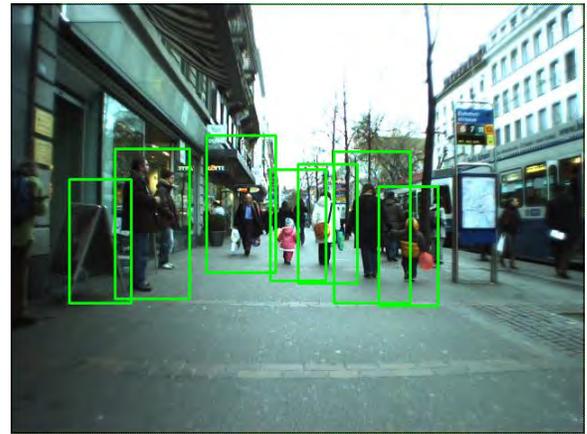


Fig. 9. Pre-trained classifier test example.

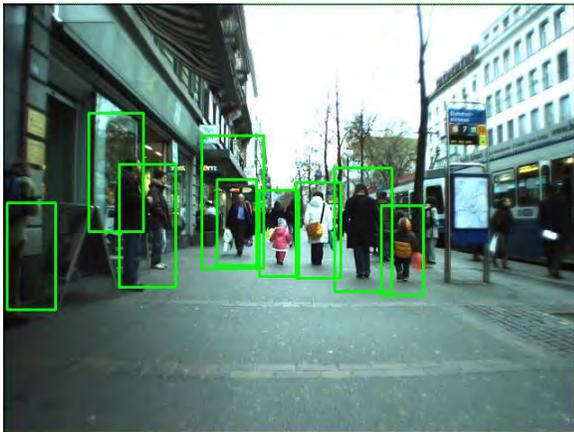


Fig. 7. Third test example.

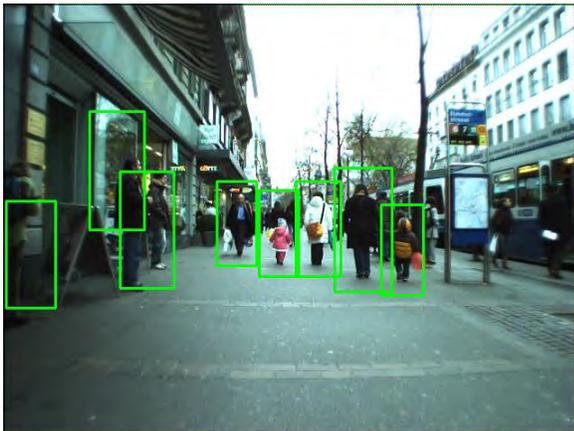


Fig. 8. Fourth test example.

C. Detection Results Curves

The curves in Fig. 10 show results of our five experiments, the curves in Fig. 11 show the comparison of our best result and the results in the paper^[1]. The results in the paper are better than our experimental results, we also need to consider some other effects, such as the difference of test samples.

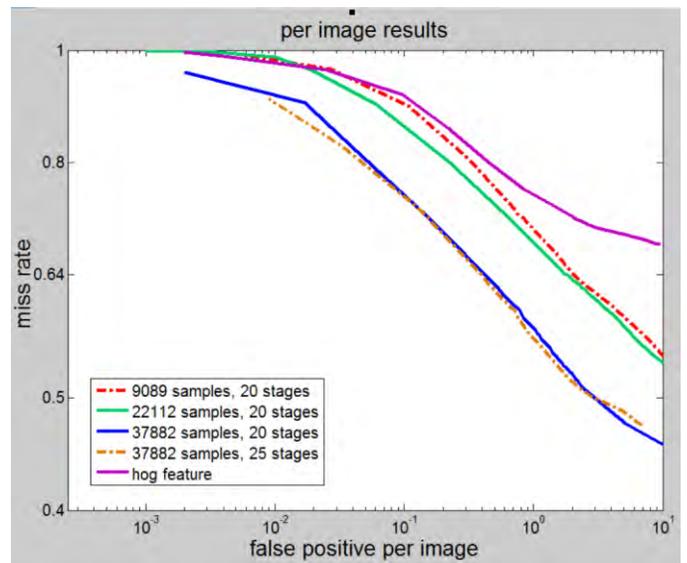


Fig. 10. Per image results of our five results.

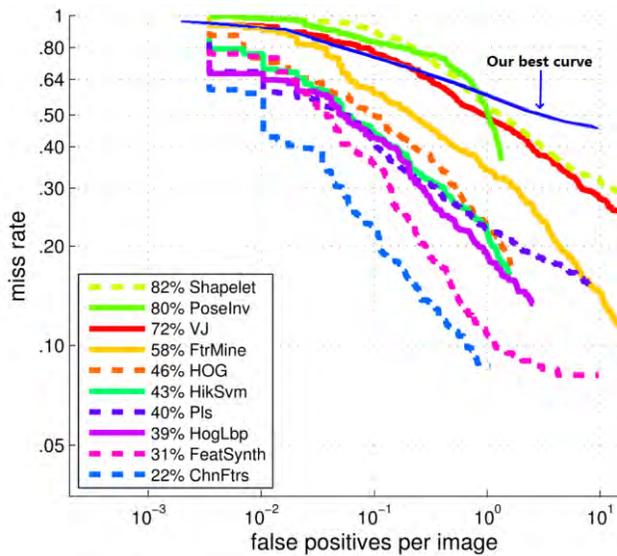


Fig. 11. The comparison of our best result and the results in the paper.

V. CONCLUSION

This paper describes our work of pedestrian detection from database, methods and experiments. In recent 10 years, pedestrian detection and computer vision developed rapidly, various methods had been proposed to improve the results. We used several easiest methods to detect pedestrian in the complex environment.

After study, we got some results in computer pedestrian detection. During this process, we did lots of experiments and trained couple of classifiers. More samples, better results. Actually, the accuracy of experiments still has enough space for further improvement.

REFERENCES

- [1] Dollar P, Wojek C, Schiele B, et al. Pedestrian Detection: An Evaluation Of The State Of The Art[J]. Pattern Analysis & Machine Intelligence IEEE Transactions on, 2012, 34(4):743-761.
- [2] Dalal N, Triggs B. Histograms of oriented gradients for human detection[J]. Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, 1(12):886-893.

Carnegie Mellon
THE ROBOTICS INSTITUTE
<http://riss.ri.cmu.edu/>