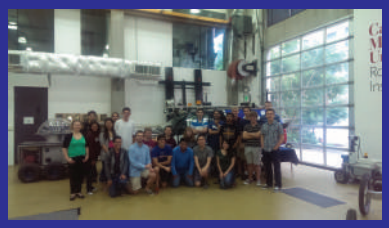
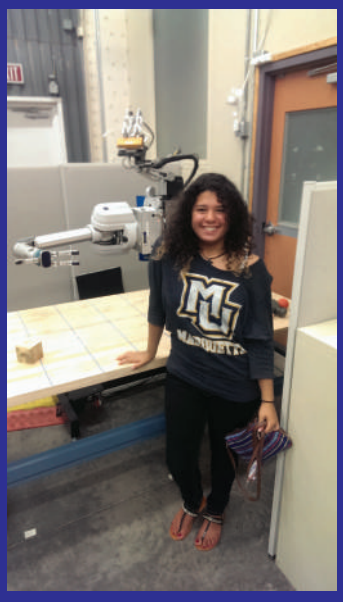
VOLUME 2 FALL 2014

ROBOTICS INSTITUTE
Summer Scholars (RISS) Working Papers

JOURNAL





Robotics Institute Summer Scholars Working Papers Journal

Volume 2 Fall 2014

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.

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.

Table of Contents

Table of Illustrations

Pictures used in “To Future Summer Scholars” can be found at

Picture 1: http://www.ri.cmu.edu/ri_static_content.html?menu_id=465

Picture 2: http://www.ri.cmu.edu/ri_static_content.html?menu_id=465

Picture 3: https://www.flickr.com/photos/126261794@N02/15022925376/

Picture 4: http://www.tartanracing.org/hires/01.jpg

Pictures used in “Photo Gallery”

Courtesy of Debra Tobin

Pictures of Students used in “Articles”

Courtesy of Debra Tobin

Congratulations Cohort of 2014!

M. Talha Agcayazi	George Mason University
Tiffany (Miki) Bassey	Carnegie Mellon University
Deanna Biesan	Baldwin Wallace College
Sabih Bin Wasi	Carnegie Mellon University in Qatar
Macauley S. Breault	Muhlenberg College
Zhe Cao	Wuhan University
Shushman Choudhury	Indian Institute of Technology, Kharagpur
Micah Corah	Rensselaer Polytechnic Institute
Christopher Eriksen	Harvey Mudd College
Nanxuan Huang	Nanjing University of Science and Technology
Li Liu	Xi'an Jiaotong University
Kenneth Marino	Georgia Institute of Technology
Pedro Mediano	University of Valencia
Mrinal Mohit	Indian Institute of Technology, Kharagpur
Kristina Monakhova	State University of New York, Buffalo
Cristina M. Morales Mojica	University of Puerto Rico at Bayamon
Andrew Orekhov	University of Tennessee
Gene Temple Price	Swarthmore College
Zhuhan Qiao	Nanjing University of Science and Technology
Suryansh Saxena	Delhi Technological University
Robert Thome	Pennsylvania State University
Luis E. Valle	University of Florida
Yujun Wang	Nanjing University of Science and Technology
Benjamin Weinstein-Raun	Virginia Tech
Danfei Xu	Columbia University
Yi Yang	Xi'an Jiaotong University, China
Hesham Zaini	Massachusetts Institute of Technology
Andy Zeng	University of California, Berkeley
Sam Ling Zeng	Carnegie Mellon University

2014 NSF Research Experiences for Undergraduates RISS Scholarship Recipients

	M. Talha Agcayazi	George Mason University
	Tiffany (Miki) Bassey	Carnegie Mellon University
	Deanna Biesan	Baldwin Wallace College
	Macauley S. Breault	Muhlenberg College
	Micah Corah	Rensselaer Polytechnic Institute
	Christopher Eriksen	Harvey Mudd College
	Kristina Monakhova	State University of New York, Buffalo
	Cristina M. Morales Mojica	University of Puerto Rico at Bayamon
	Luis E. Valle	University of Florida

Thank You!

Special Thanks to the RISS Working Papers Journal Team

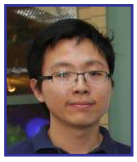
Shushman Choudhury — Indian Institute of Technology, Kharagpur

Micah Corah — Rensselaer Polytechnic Institute

Kenneth Marino — Georgia Institute of Technology

Pedro Mediano — University of Valencia

Benjamin Weinstein-Raun — Virginia Tech



Danfei Xu — Columbia University

To Future Summer Scholars

The Robotics Institute Summer Scholars Program (RISS) is a research experience that gives undergraduate students the opportunity to work on research projects in the fields of robotics and intelligent systems, supervised by faculty from the Robotics Institute at Carnegie Mellon. RISS also provides a series of events and seminars to help students prepare for graduate school and future careers in academia and industry.

The experience of a singular focus on one research project is unknown to many undergraduates, who typically divide their attention between coursework, extracurricular activities, and, if time permits, research. RISS gives students the rare experience of devoting their productive effort to one project, giving them a sense of what's involved in a career of academic research. Scholars work closely with faculty mentors and graduate students on real research projects, and make tangible contributions to that research.

Scholars are given numerous opportunities to build relationships with faculty, graduate students and other scholars, both in and outside of their individual projects

As a cohort, we have developed a close community outside of our research. Many connections and friendships we developed over the summer will persist into our careers as roboticists and researchers.

The program also helps students to prepare for the path ahead of them, including preparing for, being admitted to, and attending graduate school. RISS helps scholars by offering seminars, information sessions, and individual meetings to help them apply to graduate schools, write statements of purpose and apply to fellowships. Many scholars from

previous years have gone on to research positions at top graduate schools across the country.

But why go to grad school? Many undergrads that enroll in RISS come from programs at universities that position them well for jobs in industry. Students in graduate programs give up the salary and lifestyle their peers receive, while spending four to six stressful years scrambling for publications and funding. We feel that it is worth these drawbacks to enter the unique research environment that is the engine of innovation in America.

Research has been the basic building block of the future since the days of Francis Bacon and Louis Pasteur. The origins of the modern internet are found in collaborations between networking companies, the US government, and research institutions. It was research at Carnegie Mellon and Stanford that spawned the self-driving car, a technology that is certain to become industry standard in our lifetimes.

The unique environment of the university fosters ingenuity by promoting the collaborative growth of human knowledge.

Being a member of academia allows one to be at the forefront of this growth, and to influence its direction. Innate in all researchers is the drive to create and discover new things. All of the summer scholars from this year and years past are marked by our passion for our research and our desire to change the field of robotics or artificial intelligence. The RISS program has served as an excellent outlet for this passion.

In this journal you will see the fruits of our two brief summer at Carnegie Mellon. Some of the work presented here represents ongoing research efforts, while others were complete when we returned to our respective institutions. We hope that you will enjoy reading about what we have done.

Sincerely,

Kenneth Marino & Ben Weinstein-Raun
RISS Scholars 2014

Robotics Institute Summer Scholars Program

The Robotics Institute Summer Scholars (RISS) Program is an intensive undergraduate research program at Carnegie Mellon University. Summer scholars participate in innovative research that focuses on robotics as the intelligent connection of perception to action. As part of our commitment to undergraduate research, the institute hosts an National Science Foundation Research Experience for Undergraduates Site through the RISS program. All scholars work with faculty, post-doctoral fellows, researchers, graduate students, and fellow summer scholars from around the world to conduct research work in:

- Intelligence: including core AI technologies, motion planning, control theory, planning under uncertainty, POMDPS, game theory, and machine learning.
- Perception: including computer vision, stereo processing, understanding ladar and 3D sensing, state-estimation, and pattern recognition.
- Action: including work mechanisms, actuators, their design and control.

Previous scholars have worked on projects ranging from distributed sensing to autonomous flight through cluttered forests. Learn more about RI participating projects at www.ri.cmu.edu/summerscholars and scholar contributions to this research.

Through the program, scholars are:

(1) Immersed in a guided research process that enables them to experience the thrill of discovery and to adopt the role of scientist as one that is authentically their own;

(2) Inspired to pursue careers in robotics and related STEM fields and equipped with the skills and new knowledge to seize industry and graduate school opportunities;

(3) Challenged by the interdisciplinary nature of robotics, the complexity of the research, and the vast potential to impact and improve the world's quality of life;

(4) Supported by robust student-development programming that complements the research immersion and informs the student's post research experience trajectory; and

(6) New members of lifelong global community of researchers, entrepreneurs, and innovators that support, encourage, and enrich each other's lives.

The Robotics Institute at Carnegie Mellon University, the largest university-affiliated robotics research group in the world, offers a diverse breadth of research with an extensive range of applications; with over a hundred funded research projects. The Institute is a global leader in robotics research, education, and innovation. The Institute's experience, capacity, and faculty engagement extends unparalleled opportunities for students to be immersed in cutting-edge research while building in-demand STEM knowledge and skills.

The institute has eight years of experience hosting successful formal summer undergraduate research programs. The RI Summer Scholars program has grown to an average cohort size of 30 students and yields an impressive number of successful graduate school applications (at CMU and top universities around the world) and research position placements.

Offers of admissions to RI graduate programs received by RISS alums:

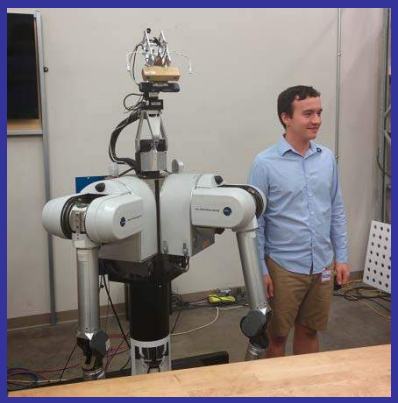
- 2012: 7 offers (1 PhD, 6 masters)
- 2013: 9 offers (4 PhD and 5 masters)
- 2014: 11 offers (4 PhD and 7 masters)

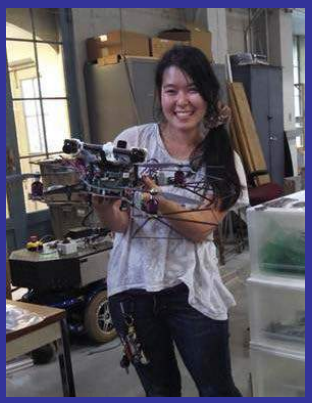
Over half of the 2014 scholars will continue to collaborate with their RISS mentors and labs. By continuing their research together and writing articles to disseminate research results, scholars have the opportunity to further develop critical skills needed for success in graduate school and industry. In addition to this 2014 RISS Working Papers Journal, at least 10 other papers are in process for submission to peer-reviewed journals and conferences. A growing number of scholars have also received offers of employment by robotics labs, centers, and companies. In 2014, three RISS alumni returned as robotics researchers to the Pittsburgh area.

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.

PHOTO GALLERY

RISS 2014





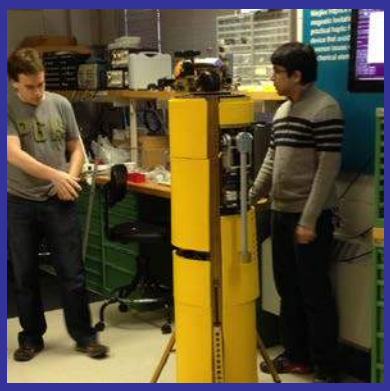
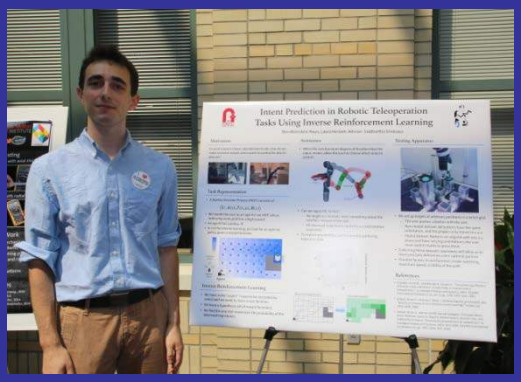
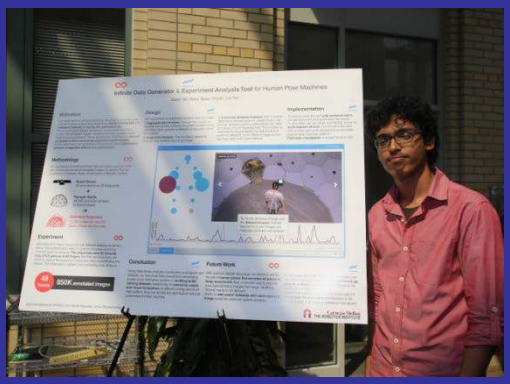
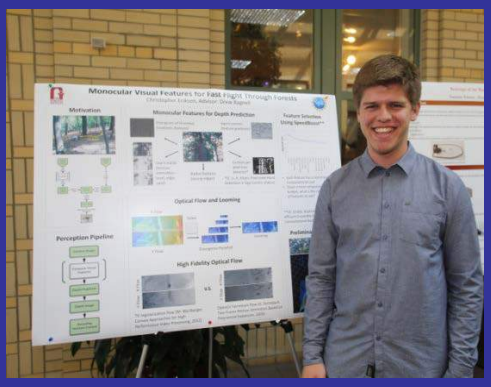
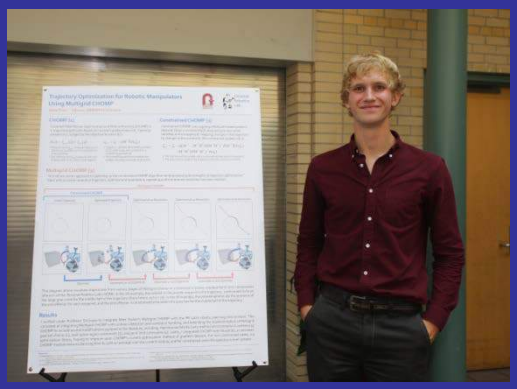


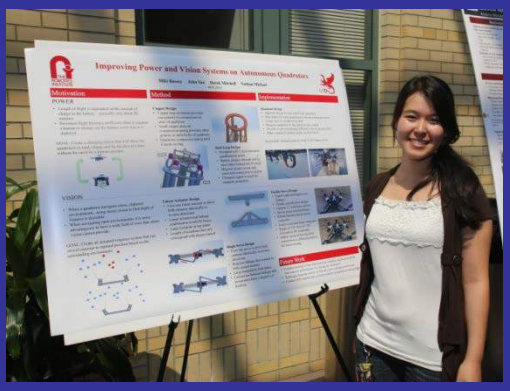
PHOTO GALLERY
RISS 2014

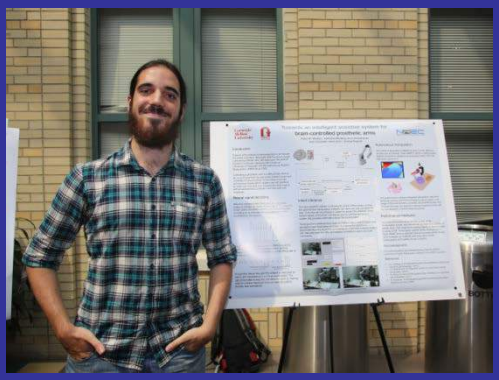


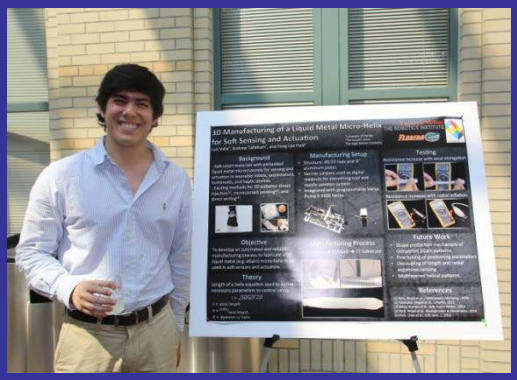












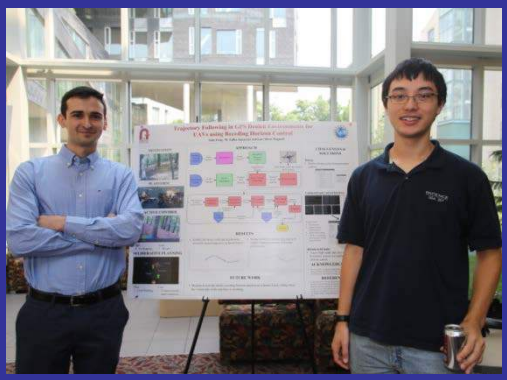


PHOTO GALLERY

RISS 2014

PHOTO GALLERY

RISS 2014
























PHOTO GALLERY

RISS 2014

Working Papers

Authors


Auto-Calibration and Hybrid Force/Position Control for the Cerberus Cardiac Robot

Macauley S. Breault, Nathan A. Wood, and Cameron N. Riviere

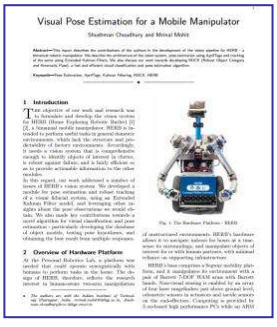
Auto-Calibration and Hybrid Force/Position Control for the Cerberus Cardiac Robot


Real-time 3D object pose estimation

Zhe Cao, Natasha Kholgade, and Yaser Sheikh

Zhe Cao, Natasha Kholgade, and Yaser Sheikh

Real-time 3D Object Pose Estimation Using Vectorized Similarity Measure


Visual Pose Estimation for a Mobile Manipulator

Shushman Choudhury and Mrinal Mohit

Shushman Choudhury and Mrinal Mohit

Visual Pose Estimation for a Mobile Manipulator


Real Time Human Pose Estimation for Boosted Random Forests and Pose Machines

Kenneth Marino, Georgia Institute of Technology

Li Liu, Vishnu R. Desaraju, Shih-Yun Lo, and Nathan Michael

Active Control of Aerodynamic Disturbance Adaptation


Rover Localization and Distance Verification Using a Planetary Landing Map

Kristina Monakhova, University at Buffalo, the State University of New York, Buffalo, New York 14260, Email: kmonakho@buffalo.edu

Maxime 'William "Red" Whittaker, Carnegie Mellon University, Pittsburgh, PA 15213, Email: red@cmu.edu

Kenneth Marino

Real Time Human Pose Estimation for Boosted Random Forests and Pose Machines


Active Control of Aerodynamic Disturbance Adaptation

Li Liu, Vishnu R. Desaraju, Shih-Yun Lo, and Nathan Michael

Kristina Monakhova and William “Red” Whittaker

Rover Localization and Distance Verification Using a Planetary Landing Map


Visual Programmer Converter for Untethered Running of the Hummingbird Duo

Cristina M. Morales Mojica, University of Puerto Rico at Rio piedras, Puerto Rico, and Dr. Illah R. Nourbakhsh, Carnegie Mellon University, Pittsburgh, Pennsylvania

Cristina M. Morales Mojica and Illah R. Nourbakhsh

Visual Programmer Converter for Untethered Running of the Hummingbird Duo


Redesign of the Waterproof Enclosure on the Lutra autonomous boat

Zhuhan Qiao

Redesign of the Waterproof Enclosure on the Lutra Autonomous Boat


System and Architecture design of NavPal outdoor navigation aid for blind and visually impaired users

Suryansh Saxena, Byung-Cheol Min, M. Bernardine Dias, and Aaron Steinfeld

System and Architecture Design of NavPal Outdoor Navigation Aid for Blind and Visually Impaired Users


Platypus Cooperate Robotic Watercraft Platform
Electronic Speed Control, Offline System, Failsafe and Auto-data Feedback

Yujun Wang, Naxuan Huang, and Yi Yang

Platypus Cooperate Robotic Watercraft Platform. Electronic Speed Control, Offline System, Failsafe and Auto-data Feedback

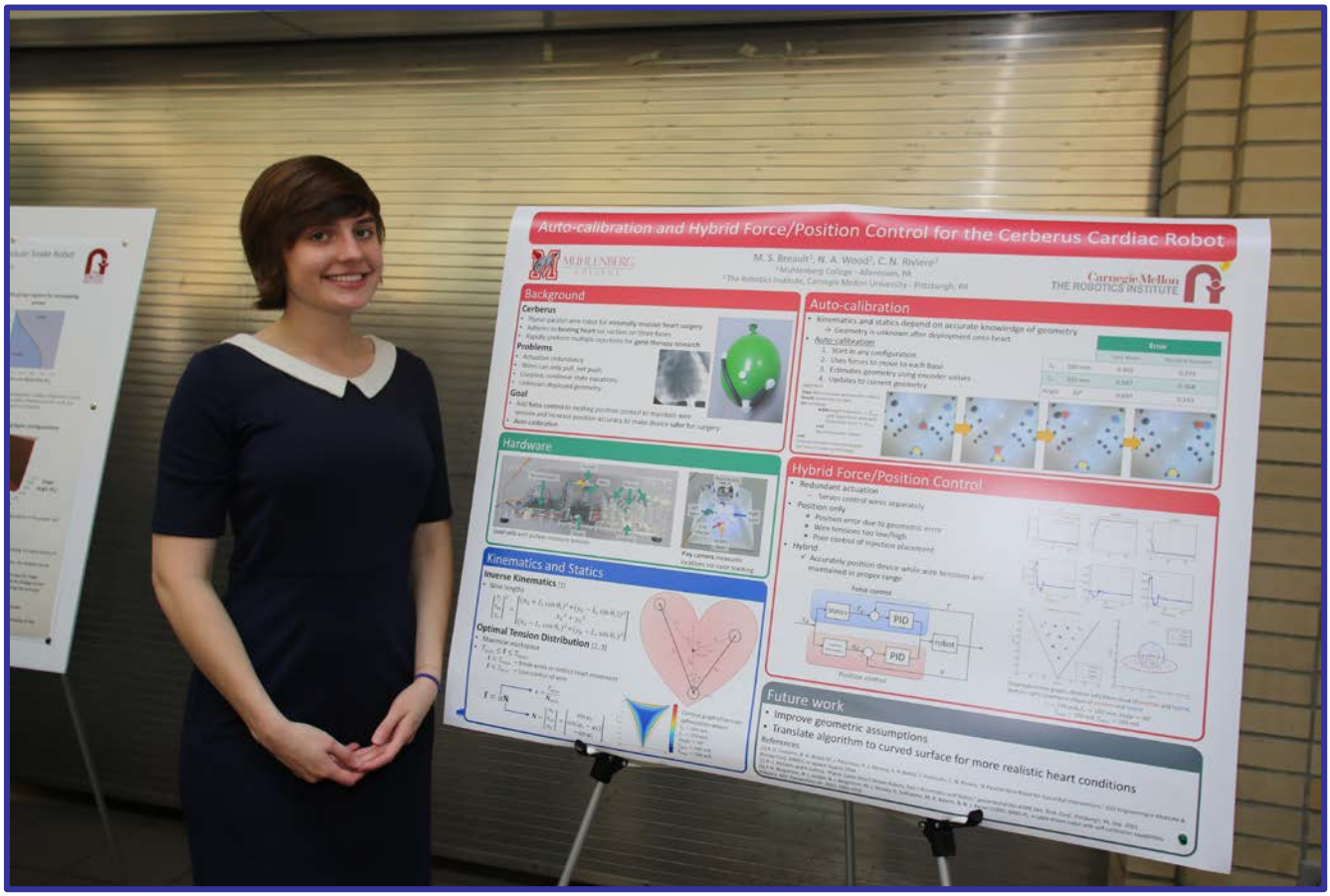

Face Alignment Refinement

Andy Zeng, Vishnu Naresh Boddeti, Kris M. Kitani, and Takeo Kanade

Face Alignment Refinement

Macauley S. Breault

RISS 2014



Auto-Calibration and Hybrid Force/Position Control for the Cerberus Cardiac Robot

Macauley S. Breault, Nathan A. Wood, Cameron N. Riviere

***Abstract* — Gene therapies have emerged as a promising treatment for congestive heart failure, yet they lack a method for minimally invasive, uniform delivery. To address this need we developed Cerberus, a minimally invasive parallel wire robot for cardiac interventions. Prior work on Cerberus was limited to controlling the device using only position feedback. In order to ensure safety for both the patient and the device, as well as to improve the performance of the device, this paper presents work on enhancing the existing system with force feedback capabilities. By modelling the statics of the system and developing a tension distribution optimization technique, existing position control schemes were modified to a hybrid force/position controller. The addition of force control was utilized in an auto-calibration procedure to measure the geometry of the robot. The presented auto-calibration routine is able to identify the shape of the device to within 0.5 mm and 0.9°, while the hybrid control scheme yields a positioning error of 1.78 mm.**

I. INTRODUCTION

A promising topic in the field of cardiovascular research has been the use of gene therapies for congestive heart failure. Current practices lack effective ways to deliver a uniform distribution of gene expression that is required for myocardium interventions [1]. Ideally, this would entail a large number of small injections that cover large areas of a beating heart where accuracy matters.

Traditional cardiac procedures involve opening the chest cavity to gain access to the paused heart and lungs. This exposes the patient to a high risk of infection and long traumatic recovery times [2]. Minimally invasive thorascopic techniques allow surgeons to reach the beating heart using rigid tools that are inserted between the ribs via small incisions [3]. Thoracic procedures are limited by the trauma inflicted by deflating the left lung in order to reveal the heart, the need to stabilize the beating heart, and the rigidity of the tools that limits the workspace. Neither option provides an effective way for the delivery of gene therapy drugs.

Cerberus is a planar parallel wire robot developed for minimally invasive cardiac interventions [4]. The device is inserted using a subxiphoid approach that accesses the heart while avoiding the lungs. Flexible arms then allow the device to expand into a triangular shape and adhere to the surface of the beating heart with suction on its three bases, providing a stable platform with no motion relative to the heart. Wires from each base connect to an injector head that moves within the triangular support structure by changing the wire lengths. This design has the typical advantages of parallel wire robots, namely a large workspace and the ability to move quickly within this workspace [5]. These advantages give the device the potential to deliver multiple injections accurately within the entirety of the workspace to the beating heart.

Previous work on Cerberus has focused on adapting previously developed methods for parallel cable manipulators to our system [4]. Under simplifying assumptions about the geometry of the robot and neglecting the curvature of the heart, inverse kinematics that yield the wire lengths were successfully derived, and a control system was developed and tested *in vivo* using only position feedback.

Errors in the calculations can cause wires to be too loose, resulting in loss of injector control and accuracy, or too tight, potentially interfering with heart activity or breaking the device. Such errors are amplified by the fact that the kinematics require accurate knowledge of the geometry which becomes altered once deployed on the heart to the point where position feedback alone will not suffice.

With the long-term goal of Cerberus to be in the operating room, it is crucial that the forces produced by the robot are monitored and controlled to ensure safety. Such forces can be measured by the tensions in the wires under the assumption that the device is frictionless and non-inertial. Further, wires can only exert force by pulling [5, 6]. Due to the device's actuator redundancy, the state equations for the forces in static equilibrium are coupled and nonlinear, leading to an infinite number of possible tension combinations. Hence, at a given point, the tension for each wire must be found by maximizing the number of wires that are within a safe range in the workspace. Limited work exists on finding tension distribution for planar cable-driven robots. While other parallel cable robots, such as NIST ROBOCRANCE, have the advantage of gravity to keep wires taut, Cerberus relies entirely on its actuators to maintain tensions [7].

In this paper, state equations for statics are adapted from previously developed methods for one degree of actuation redundancy to fit this system [5], a method to find the optimal tension distribution at a given point is developed [6], and an algorithm that estimates the geometry of the device based on force and encoder values is constructed which allows the injector to start anywhere once deployed and move independent of location knowledge. Preliminary work is also done in adding force control to the existing position control that would confine tensions within an allowable range and increase position accuracy to make the device safer for surgery.

II. METHODS

A. System Hardware

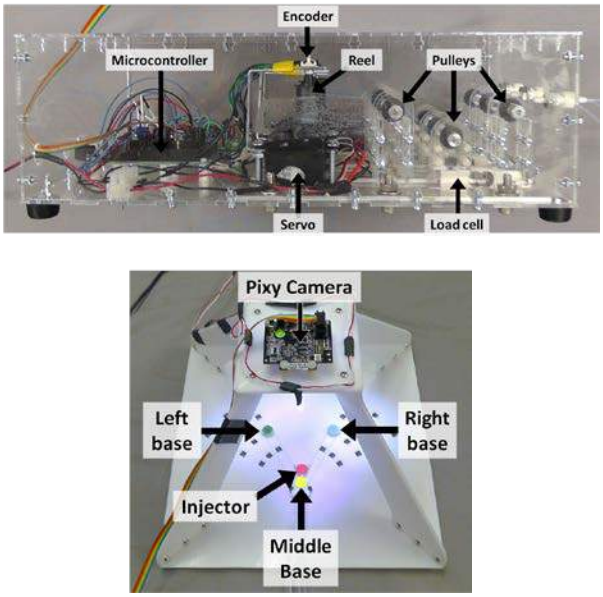


Fig. 1 (Top) Control system in acrylic box. Each servo rotates a reel that tensions the corresponding wire. The wires extend across pulleys to a corresponding load cell. (Bottom) Desktop set-up with three bases fixed on a platform, where the left and right base can be set to various lengths, and a mounted camera.

Existing electrical control system introduced in Costanza et al. [4] was adapted to fit three load cells using a pulley system and calibrated to measure the tension in each wire. A profile view of the system can be seen in Fig. 1. For the purposes of this experiment, a desktop set-up was designed capable of fixing the three bases of the robot to a planar surface while allowing variation the lengths and angles of the arms at known values as shown in Fig. 1. A Pixy camera was mounted directly above to capture all possible configurations within the camera's field of view. Ground truth was established using the camera's color tracking software via markers on the bases and injector.

B. *Kinematics*

Kinematic equations were adapted to fit this system from previous work for general parallel wire manipulators with one degree of actuation redundancy [5]. No closed form solution exists for the forward kinematics because the system is a parallel manipulator. Inverse kinematics can be found by drawing concentric circles around each base that intersect at the injector to find the Euclidean distance between each base assuming that the middle base is set as the origin under Cartesian coordinates (Fig. 2). As provided by Costanza et al. [4], the lengths of the wires are given by:

$$\begin{bmatrix} {r_l}^2 \\ {r_m}^2 \\ {r_r}^2 \end{bmatrix} = \begin{bmatrix} (x_0 + L_l \cos\theta_l)^2 + (y_0 - L_l \sin\theta_l)^2 \\ {x_0}^2 + {y_0}^2 \\ (x_0 - L_r \cos\theta_r)^2 + (y_0 - L_r \sin\theta_r)^2 \end{bmatrix} \quad (1)$$

Note that this depends on knowledge of desired injector position and the geometry of the robot. However, geometric measurements become skewed once the device is deployed, rendering the previous equations inaccurate. Upcoming sections will discuss a solution to this dilemma using an auto-calibration procedure.

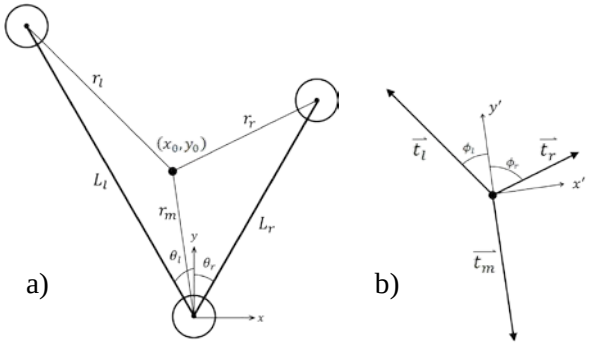


Fig. 2 a) Ideal kinematics and b) the free body diagram of the planar wire robot manipulator.

C. *Statics*

Previously developed methods by Williams et al. [5] for parallel wire robots with one degree of actuation redundancy were adapted for Cerberus. In this system, it is assumed that the mass of the end-effector, or the injector, is negligible. Then, even in motion, the system can be modeled such that the sum of the wire tensions will always be zero. A free-body diagram of this static model is shown in Fig. 2. The Cartesian coordinates are altered from the frame used in kinematics (x, y) with the middle base as the origin to (x', y'), where the injector is now taken as origin such that the middle wire lines up with negative y' with a similar transformation for the angles (Fig. 2b). Resulting static equations are given by:

$$\sum F_{x'} = t_l \sin\phi_l + t_m \sin\phi_m + t_r \sin\phi_r = 0 \quad (2a)$$
$$\sum F_{y'} = t_l \cos\phi_l + t_m \cos\phi_m + t_r \cos\phi = 0 \quad (2b)$$

where t_l, t_m, t_r are the tensions applied by the left, middle, and right wires respectively. This can also be expressed as:

$$\mathbf{S}\,\mathbf{T} = \mathbf{0}$$
$$\mathbf{S} = \begin{bmatrix} \sin\phi_l & \sin\phi_m & \sin\phi_r \\ \cos\phi_l & \cos\phi_m & \cos\phi_r \end{bmatrix} \quad (3)$$
$$\mathbf{T} = \begin{bmatrix} t_l \\ t_m \\ t_r \end{bmatrix}$$

Due to the actuation redundancy, (3) is underconstrained meaning there are infinitely many tensions combinations possible that would satisfy the equation.

D. *Optimal Tension Distribution*

In order to solve for the tensions in the system, **T**, (3) must be inverted. Then the tensions of the wires in our system can be expressed as:

$$\mathbf{T} = \alpha\mathbf{N}, \quad (4)$$

where $\mathbf{N} = [n_l \quad n_m \quad n_r]^T$ is the kernel vector of the components of the tensions from (3) and α is a scalar

weight. A method to calculate the kernel vector was adapted from Williams et al. [5] where each component is found by taking the determinant of the 2x2 submatrix of **S** with the corresponding column removed. Using this method the kernel vector for this system would be:

$$\mathbf{N} = \begin{bmatrix} n_l \\ n_m \\ n_r \end{bmatrix} = \begin{bmatrix} \sin(\phi_m - \phi_r) \\ \sin(\phi_r - \phi_l) \\ \sin(\phi_l - \phi_m) \end{bmatrix} \quad (5)$$

By setting the coordinate system such that $\phi_m = 180°$, equation (5) can be simplified to:

$$\mathbf{N} = \begin{bmatrix} \sin(\phi_r) \\ \sin(\phi_r - \phi_l) \\ -\sin(\phi_l) \end{bmatrix} \quad (6)$$

The kernel vector given by equation (6) gives us the ratios of tensions that satisfy equation (3). Recall that parallel wire robots can only exert tension. Thus, the tensions in the wires **T** will always be positive in the workspace. The signs of the tensions are expressed by the trigonometric expressions in **N**, where a point outside of the workspace will result in an angle that produces a negative n and a point taken on the edge of the workspace will result in $n = 0$.

The tensions are also constrained by $T_{min} \leq \mathbf{T} \leq T_{max}$, where $T_{min} > \mathbf{T}$ will result in a loose wire that could get caught on something and $\mathbf{T} > T_{max}$ may snap the wire, break the robot or restrict heart movement [6]. To ensure that this is satisfied by each of the wires, the scalar α must be chosen such that they all meet the minimum tension requirement for any given point within the workspace. This can be generalized as:

$$\alpha = \frac{T_{min}}{\min \mathbf{N}} \quad (7)$$

Combining (4), (6), and (7), the optimal tensions required to keep the injector static can be found for any point within the workspace. Note that tensions above the maximum allowed force must be manually rejected.

E. Auto-Calibration

Due to the flexible arms, the geometry of the robot after it is deployed onto the heart can vary. Obtaining visual confirmation of robot using medical imaging would be undesirable as it would take extra time and be more expensive. Other research has explored alternative methods of calculating cable lengths from geometric properties [6]. The method previously used by Cerberus involved manually adjusting wire lengths until the user felt that the injector was at a base given that that base's wire was fully taut. The encoder values at said base were then recorded and the user would then manually position the injector to the next base. The lengths of the robots arms were then determined using these wire lengths.

In an effort to automate this process, an auto-calibration routine was developed that takes advantage of force measurements. Similar to the manual process, the auto-calibration begins by pulling the wire of a designated base as the other two wires maintain a minimum tension as to not inhibit the main wire. The robot will know that it is at the base when its designated wire cannot be pulled in without exceeding the maximum while the other wires are within the minimum range. This process repeated to record the encoder values at each base. The encoder values are then converted into wire lengths based on the reel's diameter where the difference between each base will give the designated arm length as well as the angle between the left and right base. A summary of this algorithm can be seen in Fig. 3. Accuracy and precision of the auto-calibration algorithm was tested by running it under varying the geometry of the set up and comparing the results to the known geometry values.

Data: Wire tensions and encoder values
Result: Geometry of robot
for each base
  while target base wire $< T_{max}$
    pull target base wire and
    keep other wire $\approx T_{min}$
  end
  Record encoder values
end
Convert encoder values to lengths
Use law of cosine to find angle

Fig. 3 Algorithm for auto-calibration.

F. Hybrid Force/Position Control

Due to the device's unique design, the actuation redundancy means that each wire must be controlled separately via servos. Prior controls relied on varying individual wire lengths, calculated using inverse kinematics based on the desired position and the encoder values of the servos translated into lengths, to move the injector. Hence errors in calculation or geometric error could cause errors in position allowed for the possibility of loose or too tight wires. Such occurrence would translate to the surgeon having poor control of injection placement in an application setting.

With the ability to monitor tensions in the wires, the possibility of melding both position and force measurements to control the injector's movement and location is explored. A hybrid of force and position control would give the best of both scenarios; allowing for accurate position of the injector, determined by wire length and optimal tensions, while wire tensions are maintained within the proper range. A parallel controller was implemented using two PID controllers using desired location. One controller used force feedback as the input with the set point based on the calculated desired tensions for the desired target location. The other controller used position feedback as the input with the set point based on the calculated desired wire lengths for the desired target location. Both yield portions of the servo speed that, when added together, resulted in the actual speed. In cases where the injector was far from the target but a tension was too high, the corresponding servo would release wire quicker.

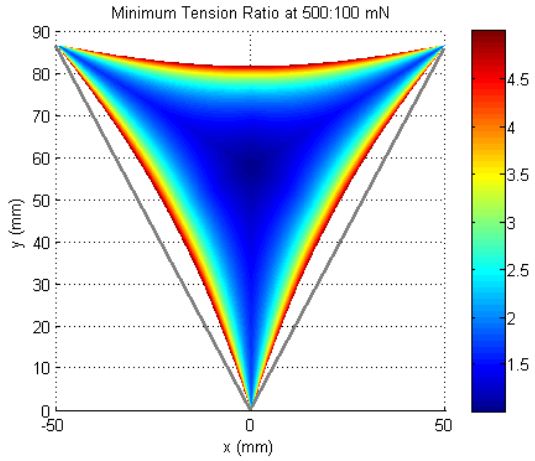


Fig. 4 Contour graph of tension optimization under the geometry of $L_l = 100\ mm$, $L_r = 100\ mm$, $\theta = 30°$, $T_{min} = 100\ mN$, and $T_{max} = 500\ mN$.

The hybrid force/position controller was tested against the controller that used only position feedback for various geometric configurations. The injector was sent to ten desired locations within the workspace. These points were compared to the injectors' actual location obtained from a camera via color markers. The resulting points were compared to the desired locations to find error and covariance.

III. RESULTS

Simulations and bench top experiments were conducted to verify the validity of tensions measurements, auto-calibration results and hybrid controller under operation conditions.

A. *Optimal Tension Distribution*

A method of calculating the tension distribution from desired injector location such that all tensions are at least the minimum tension requirement for every point in the workspace was found. A model of the device was simulated using MATLAB to assess the dependence of the workspace with regard to geometry and tension limits.

The simulation was able to calculate the optimal tension distributions with various arm lengths and angles, validating that the technique is robust against any geometry. An example of such a distribution is demonstrated in Fig. 4, where the color map indicates the minimal tension required by the wires within the workspace as a ratio between the set maximum and minimum tensions. Smallest tensions closer to the minimum tension are shown in blue while the smallest tensions closer to the maximum allowed tension are shown in red. The lowest minimum tensions are in the middle of the workspace where as the highest minimum tensions are present on the boundaries of the workspace. The lack of color around the arms of the robot indicates that tensions in those areas could not satisfy the constraint, namely that the larger tensions were above the maximum. The amount of usable workspace depends on the ratio of maximum to minimum tension. Increasing the maximum tension will increase the accessible workspace. Conversely, this technique can be used to determine the geometry that optimizes the amount of usable workspace.

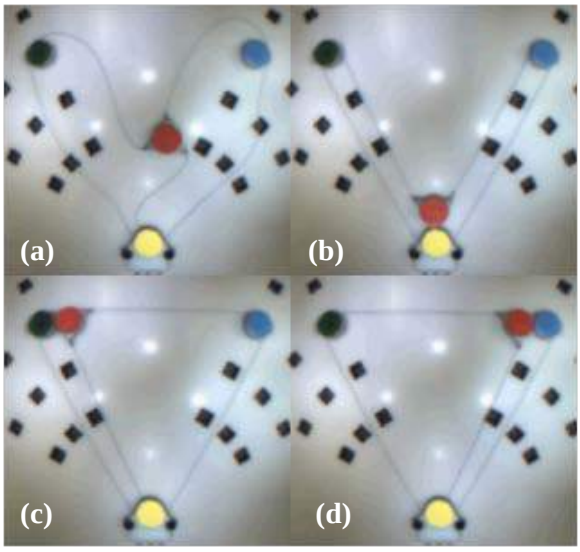


Fig. 5 Screenshots during auto-calibration (a)-(d). (a) Robot starts in any configuration. (b) Injector moves to middle base. (c) Injector moves to left base. (d) Injector moves to right base and then restarts back at middle base.

B. *Auto-Calibration*

Kinematics and statics depend on accurate knowledge of the arm lengths and angles of Cerberus to position the injector. This geometry is unknown once robot is deployed onto the heart. An auto-calibration algorithm was designed that can estimate the unknown geometry of the device after it is deployed onto the heart and allows the injector to start in any arrangement. An example of the auto-calibration process in screenshots is shown in Fig. 5.

Auto-calibration was tested using various geometries to verify the accuracy of the procedure. Estimates of the left and right arm lengths and symmetric angle between the arms compared from the calibration procedure were compared to the actual measurements. The average mean error of the left and right arm were 0.47 mm (±0.37) and 0.59 mm (±0.30) where the actual dimensions were both 100 mm while the mean error for the angle was 0.90° (±0.14) with the actual angle being 30°.

C. *Hybrid Force/Position Control*

A preliminary controller was implemented that uses desired wire lengths and tensions to move the injector. These controls were used in parallel with a PID controller and combined to output the speed of the servo to control the wire lengths. Fig. 6 shows the time trial results of hybrid controller of desired values and actual values showing each wire achieving desired tension within 20 seconds. To reach this tension, the wires had to deviate away from its desired length value. This compromise is present due to the small inaccuracies in the calculation for the wire length due to oversimplification of the system.

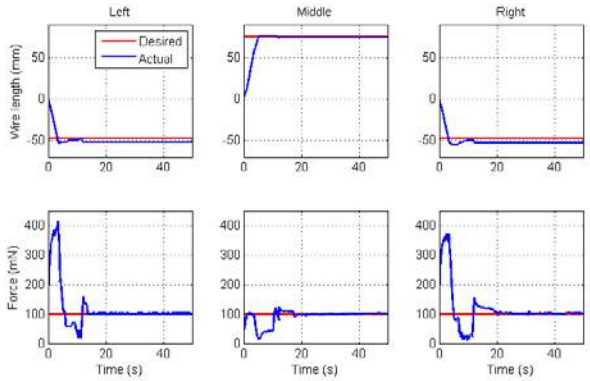


Fig. 6 Experimental arm lengths and tension distribution results when injector commanded to go to center of workspace for one sample run. Commanded values are shown in red and experimental measurements are shown in blue. Assume injector starts at middle base where the left and right wires are fully extended. Negative wire lengths indicate that the wire must retract.

Movements of the injector were divided into stages. After a desired position is commanded via the surgeon, the robot calculates the desired wire lengths and optimal tensions. In the first stage of the motion, both position and force control are given equal weights in moving the robot. As the injector closes in one the desired wire lengths within 1 cm of the desired wire lengths, position control is weighted heavier than force control until the wire lengths are within a 2 millimeters of desired values. After which, force control takes over and alters the wire lengths until the desired statics are achieved. This can be seen in Fig. 6 where within 10 seconds after starting, the arm lengths diverge from their desired values to compensate for tensions.

The injector's position accuracy was tested using hybrid force/position control and position-only control. The injector was commanded to go to ten known coordinates within the workspace. Ground truth of the end-effectors was found using the built-in color tracking system of a Pixy camera fixed above the platform set-up. Arm lengths and angles were varied. An example of an experiment is seen in Fig. 7. Local accuracy of the hybrid approach appears to be no better than position-only control with improvements in some areas. Analysis of global accuracy reveals that hybrid control has proportional error for both x and y position error while position control has significantly higher error in the x-direction. Overall, the hybrid controller yielded of positioning error of 1.78 ± 0.78 mm, while the position control yielded an error of 2.89 ± 1.40 mm

IV. DISCUSSION

Current models of the device are simplified for ideal conditions. Future work could focus on improving the geometric assumptions made to better reflect realistic conditions. Particularly taking into account the base diameter will change both wire length and angles used to calculate optimal tensions. Further work could also explore different controller architectures. The ideal controller would be able to balance the relationship between force and position for smoother movements without stages. *In vivo*

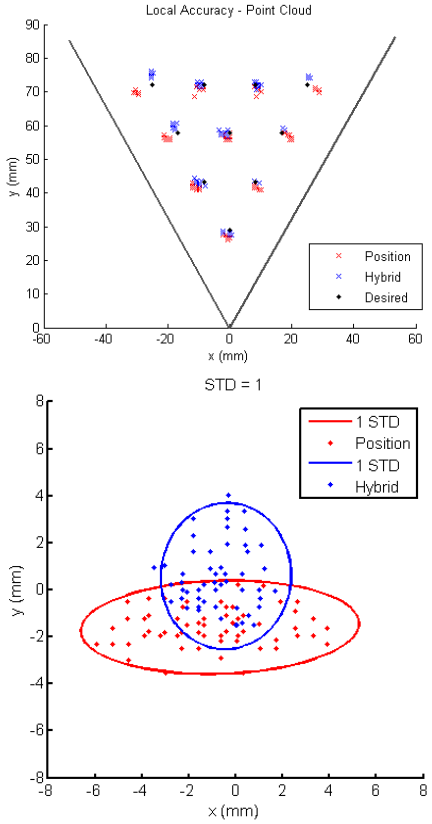


Fig. 5 Local (left) and global (right) point positioning results from 10 points (black) within workspace using hybrid (blue) or position (red) controller with the geometry of $L_l = 100\ mm$, $L_r = 100\ mm$, $\theta = 30°$, $T_{min} = 100\ mN$ and $T_{max} = 500\ mN$.

should be performed to assess how the periodic motion of the heart will affect the additional force control.

The next step for this robot would be to translate the algorithm from the current planar model to a curved surface to replicate more realistic heart conditions. This would begin on an idealized sphere and then move onto irregular curved surfaces similar to the heart. Finally, the movement of the heart should be taken into account so that the wires will adjust to move with the beating heart as to maintain constant tensions in the wire without moving the injector.

V. CONCLUSION

Force measurements of a parallel wire manipulator with actuation redundancy for cardiac interventions were successfully integrated with the pre-existing system using load cells. The capability of monitoring tensions of wires ensures the safety of the patient by ascertaining that the forces are within a range that allows the surgeon to maintain control of the injector's placement by keeping wires taut yet low enough to maintain the integrity of the robot as well as to not interfere with heart movement. The additional information about the system's state also means that desired injector position potentially can be more accurate. More importantly, the injector can now be moved without knowledge of position, allowing for the ability to update the geometry of the robot after it is deployed using auto-

calibration. This geometry is essential part of calculating the wire lengths and tensions that are used to move the injector.

REFERENCES

[1] M. Hedman, J. Hartikainen, and S. Ylä-Herttuala, "Progress and prospects: Hurdles to cardiovascular gene therapy trials," *Gene Ther.*, vol. 18, pp. 743-749, 2011.

[2] M. J. Mack, "Minimally invasive and robotic surgery," *JAMA*, vol. 285, no. 5, pp. 568-572, 2001.

[3] M. J. Mack, "Minimally invasive cardiac surgery," *Surg. Endosc.*, vol. 20, no. Suppl 2, pp. S488-92. Apr. 2006.

[4] A. D. Costanza, N. A. Wood, M. J. Passineau, R. J. Moraca, S. H. Bailey, T. Yoshizumi, C. N. Riviere, "A Parallel Wire Robot for Epicardial Interventions," *IEEE Engineering in Medicine & Biology Conf. (EMBC)*, pp. 6155-8, Aug. 2014.

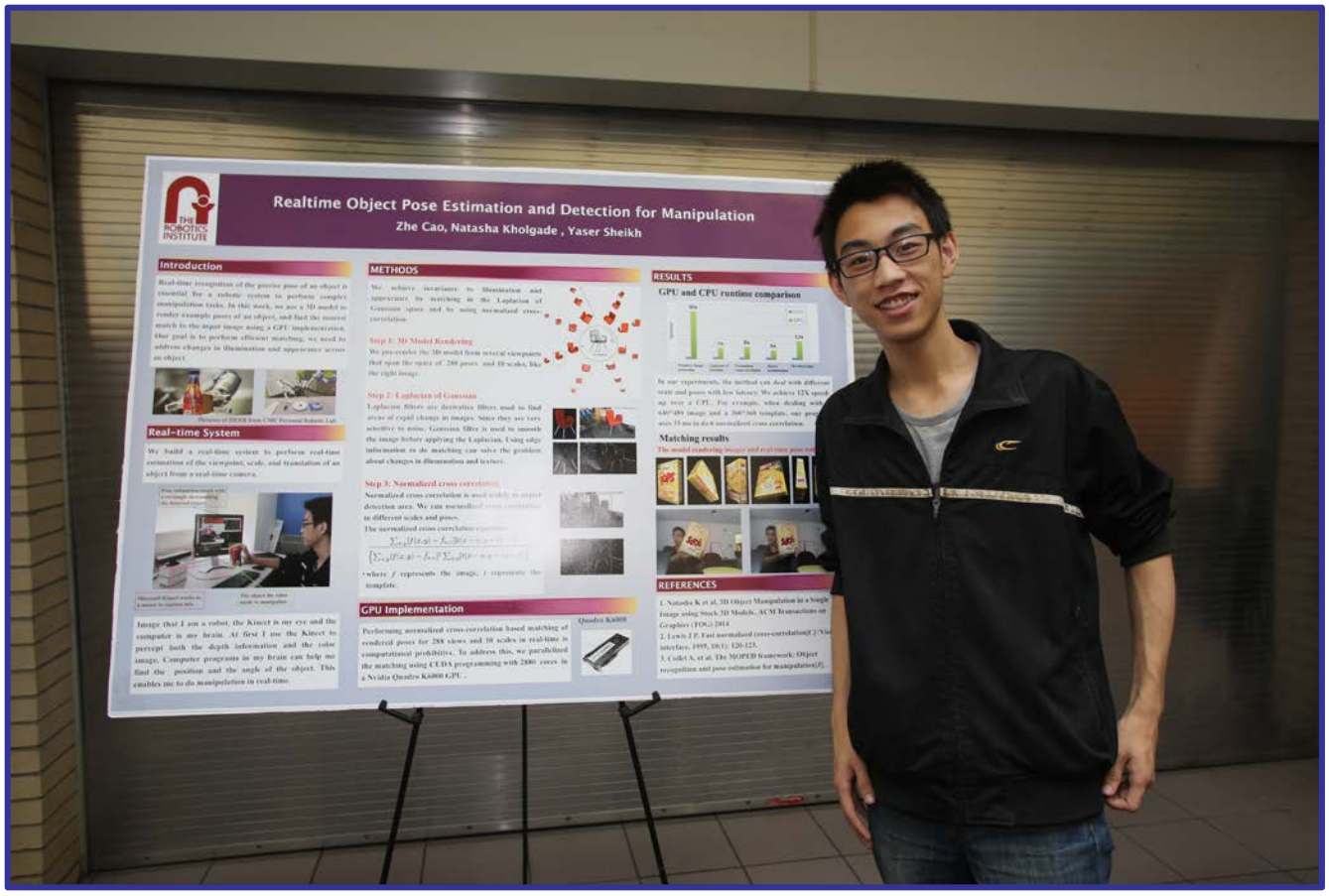
[5] R. L. Williams and P. Gallina, "Planar Cable-Direct-Driven Robots, Part I: Kinematics and Statics," *presented at the ASME Des. Tech. Conf.*, Pittsburgh, PA, Sep. 2001.

[6] P. H. Borgstrom , B. L. Jordan, B. J. Borgstrom, M. J. Stealey, G. Sukhatme, M. A. Batalin, and W. J. Kaiser. "NIMS-PL: a cable-driven robot with self-calibration capabilities." *IEEE Transactions on* Robotics, vol. 25, no. 5, pp. 1005-10, Oct. 2009.

[7] J. Albus, R. Bostelman, and N. Dagalakis, "The NIST ROBOCRANE," *J. Robot. Syst.*, vol. 10, no. 5, pp. 709-24, 1993.

Zhe Cao

RISS 2014



Real-time 3D Object Pose Estimation Using Vectorized Similarity Measure

Zhe Cao zhecao@andrew.cmu.edu
Natasha Kholgade nkholgad@cs.cmu.edu
Yaser Sheikh yaser@cs.cmu.edu

Abstract

We present a detection-based approach to perform real-time estimation of the viewpoint, scale, and translation of an object in front of the camera. In this work, we use a 3D model to render example poses of an object, and find the nearest match to the input image using a GPU implementation. To achieve invariance to illumination and appearance across an object, we transform images to the Laplacian of Gaussian space. To meet the real-time requirement, we restructure both template set and the image into two huge matrices and simultaneously process hundreds of templates with the vectorized normalized cross correlation. We further speed up our method by PCA based matrix dimension reduction and candidate elimination method. Comparative results for several image sequences are shown to validate the effectiveness of our approach.

1. Introduction

Real-time recognition of the precise pose of an object is an essential function for many computer vision applications, like robotic manipulation, scene understanding and human-computer interaction. Though impressive results have been produced, these methods usually depend on long-time offline training stage or various tracking methods to achieve real-time speed. We present a more general model-based pose estimation method without using tracking method. We achieve good accuracy and time performance (see Section 4). In summary, our method has the following main contributions.

(1) Present a robust pose estimation method in the photometric invariant image space. Our method address changes in illumination and appearance across an object by Laplacian of Gaussian transformation.

(2) Develop a novel GPU-based vectorized Normalized Cross-Correlation algorithm aimed at improving the performance of exhaustive template matching. We achieve state-of-art speed for template matching.

(3) Propose a PCA based candidate elimination method to limit the searching space of the image and speed up large sparse matrix multiplication .

(a) Frame (b) Result

Figure 1. Example frames from our object detection and pose estimation results. Our detection-based method does massive template matching in the laplacian of Gaussian space. It is capable of handling different illumination, both texture-less and textured objects. Our method produces multiple hypotheses, each top pose is drawn in white contour on the image with different line width.

(4) Introduce a general approach for the real-time pose estimation system with multiple hypothesis output. The same approach can be applied into depth image similarity measure. The multiple hypotheses can be integrated in the particle filtering based tracking method.

2. Related work

Real-time object detection and 6DOF pose estimation is critical for robotic applications involving object grasping and manipulation, and also human robot interaction. A popular approach to the pose estimation problem is to establish visual feature matches between an input 2D image and a 3D object model [1], [2], [3]. However, lack of texture, object appearance change or occlusion will make it difficult to find the reliable visual features in the image.

Another way to estimate an 3D object pose is massive template matching, that is, matching whole templates to an input image and finding the best match. The object templates are generated by rendering or capturing the 3D

model in different viewpoints. However, this approach has two main drawbacks: Firstly, template matching usually needs long computation time. Numerous techniques aimed at speeding up the basic approach have been devised. Dominant orientation template [6] represent an image as a grid of dominant orientation in each image patch. Other methods reduce the computation cost by organizing templates in a hierarchical structure [7], [8]. However, these techniques imply a non-exhaustive search process because they do not compare the full resolution image with each template at every search position and can be trapped by local optimum resulting in wrong localisation of the template.

Normalized cross-correlation (NCC) is another effective template matching method which makes pixel-wise comparison between the image and the template. Since it is computationally expensive, many research works have been done to speed up NCC. Fast NCC algorithm [10] was proposed to decrease the computation load by using precomputed integrals of the image and using fast fourier transform in the frequency domain. GPU based NCC has been implemented [11] [12] [13] and it becomes a new feature in Matlab 2014a. Several work intend to simplify the procedure of NCC by using an upper bound of the NCC function [15] [16] [17], or using Cauchy-Schwartz inequality [18] [19]. All these approaches need to process one template each time and cannot meet the requirement of real-time 3D pose estimation in which a large number of templates are required to cover all possible object rotations and scales.

Secondly, template matching is sensitive to difference in illumination or appearance between the template and the image. This requires the model-based matching method to make a precisely same 3D model for detection and pose estimation. Several work try to overcome this drawback by matching the images with edge pixels orientation [5] [6] or corner points orientation [14] [20].

In this paper, we perform real-time 3D object pose estimation in the Laplacian of Gaussian space via massive templates matching, one template for each pose, while addressing the computational efficiency by using GPU-based vectorized NCC and PCA based dimensionality reduction.

3. Method

3.1. Vectorized similarity measure

Given one object, we pre-render its 3D model from several viewpoints that span the space of rotations as well as several scales to get n templates. Given an image I of the object, we match all templates with the image simultaneously by vectorizing templates and image patches. Let us consider a template $T_i \in \mathbb{R}^n$, here $i \in \{1, 2, \ldots, n\}$, and n is the number of templates. We vectorize this template into a vector $\mathbf{t}_i \in \mathbb{R}^m$, m is the pixel number in one template. Similarly, consider an image patch P_j with same

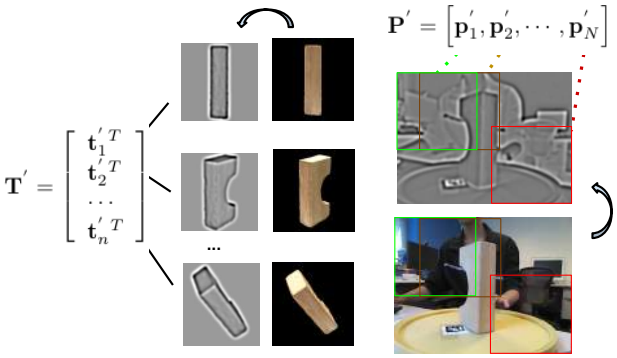


Figure 2. Image matrix $\mathbf{P}'$ and template matrix $\mathbf{T}'$.

size of the template at location j in the image $I \in \mathbb{R}^N$, where $j \in \{1, 2, \ldots, N\}$, and N represents the number of image patches in I. We vectorize this patch into a vector $\mathbf{p}(j) \in \mathbb{R}^m$. The cross-correlation between the two vectors, $\mathbf{t}_i$ and $\mathbf{p}_j$, represents the similarity between the i-th template T_i and j-th image patch P_j.

$$s = \mathbf{t}_i^T \mathbf{p}_j, \tag{1}$$

where s represents two vectors' similarity. However, simply using cross-correlation is not robust to the different illumination, different model appearance and deformation change between the template and image patch.

Our aim is to develop a robust similarity measurement method between the template vector and image patch vector. Specifically, we want to transform the patches $\mathbf{t}_i$ and $\mathbf{p}_j$ to $\mathbf{t}_i'$ and $\mathbf{p}_j'$ under a function $\mathbb{f}(\cdot) : \mathbb{R}^m \leftarrow \mathbb{R}^m$, i.e., to compute $\mathbf{t}_i' = \mathbb{f}(\mathbf{t}_i)$, and $\mathbf{p}_j' = \mathbb{f}(\mathbf{p}_j)$. Under this function, we use the cross-correlation as a robust way to measure the similarity of these two vectors. We find the method is robust to different illumination, different model appearance, texture and even slight deformation change if the transformation function $\mathbb{f}(\cdot)$ performs the mean-variance normalization of the Laplacian of Gaussian of the image patch, i.e., if

$$\mathbb{f}(\cdot) = (\mathbb{f}_{mvnorm} \circ \mathbb{f}_{LoG})(\cdot), \tag{2}$$

where

$$\mathbb{f}_{mvnorm}(\mathbf{v}) = \frac{(\mathbf{v} - \mu_{\mathbf{v}})}{\sigma_{\mathbf{v}}}, \tag{3}$$

$\mu_{\mathbf{v}}$ and σ_v being the mean and standard deviation of the intensity values of a vectorized patch $\mathbf{v} \in \mathbb{R}^m$, and

$$\mathbb{f}_{LoG}(\mathbf{u}) = (\Delta \mathbb{G}) * \mathbf{u}, \tag{4}$$

where $\mathbf{u}$ is a patch from the original image, $\mathbb{G}$ represents a two-dimensional Gaussian, and Δ represents the Laplace operator.

We then represented the similarity between the transformed patches $\mathbf{t}_i'$ and $\mathbf{p}_j'$ by:

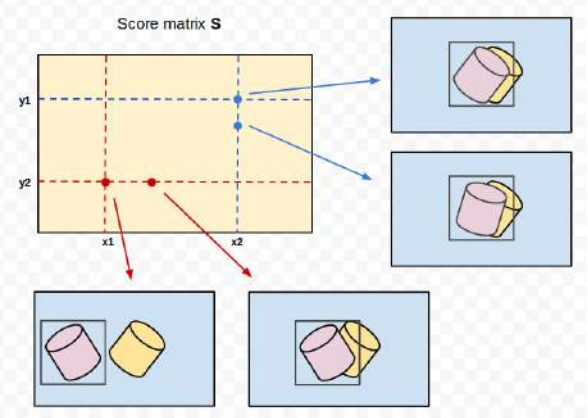


Figure 3. Illustration of the score matrix $\mathbf{S}$

$$s^{'} = \mathbf{t}_i^{'T}\mathbf{p}_j^{'}, \tag{5}$$

where $s^{'}$ represents the normalized cross-correlation between the Laplacian of Gaussians of the original patches T_i and P_j.

By vectorizing the patches, we leverage the matrix-vector multiplication on GPU to speed up the computation. We develop a score matrix, $\mathbf{S}$ with the size of $R^{n\times m}$, where n is the number of templates and m is the number of patches in the image.

$$\mathbf{S} = \mathbf{T}^{'}\mathbf{P}^{'}, \tag{6}$$

where $\mathbf{T}^{'}$ is the vectorized template matrix, each row of $\mathbf{T}^{'}$ represents one vectorized template $\mathbf{t}_i^{'}$. $\mathbf{P}^{'}$ is the image matrix, each column of $\mathbf{P}^{'}$ represents one vectorized image patch $\mathbf{p}_j^{'}$ and $\mathbf{S}(i,j)$ represents the correlation value of i-th template T_i with the j-th image patch P_j.

$$\mathbf{T}^{'} = \begin{bmatrix} \mathbf{t}_1^{'T} \\ \mathbf{t}_2^{'T} \\ \cdots \\ \mathbf{t}_n^{'T} \end{bmatrix}, \mathbf{P}^{'} = \left[\mathbf{p}_1^{'}, \mathbf{p}_2^{'}, \cdots, \mathbf{p}_N^{'}\right] \tag{7}$$

3.2. Dimensionality reduction in large sparse matrix

After vectorizing all the templates, we get one large sparse matrix $\mathbf{T}^{'}$. To further speed up calculation of equation (5), we precompute a dimensionality reduction on the template matrix $\mathbf{T}^{'}$.

$$\mathbf{T}^{'} = \mathbf{A}\mathbf{Z}, \tag{8}$$

where $\mathbf{A}$ is an orthogonal matrix and $\mathbf{Z}$ is the basis matrix with the first k principal components. To obtain $\mathbf{A}$ and $\mathbf{Z}$, we do the singular value decomposition on $\mathbf{T}^{'}$:

$$\mathbf{T}^{'} = \mathbf{U}\mathbf{D}\mathbf{V}^{T}, \tag{9}$$

where $\mathbf{D}$ represents a diagonal matrix of the same dimension as $\mathbf{X}$, with non-negative diagonal elements in decreasing order, and $\mathbf{U}$ and $\mathbf{V}$ are diagonal matrices unitary matrices. Here we select k principal components and keep α percentage variance:

$$\frac{\sum_{i=1}^{k}\mathbf{D}_{ii}}{\sum_{i=1}^{n}\mathbf{D}_{ii}} \geq \alpha. \tag{10}$$

In our experiments, we find α = 90% can keep the accuracy while reducing dimensionaloty of the matrix.

If we select first k columns from $\mathbf{U}$

$$\mathbf{A} = [\mathbf{U}_1, \mathbf{U}_2 \ldots \mathbf{U}_k], \tag{11}$$

and define $\mathbf{Z}$ as

$$\mathbf{Z} = \begin{bmatrix} \mathbf{D}_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{D}_{kk} \end{bmatrix} \begin{bmatrix} \mathbf{V}_{11} & \cdots & \mathbf{V}_{mk} \\ \vdots & \ddots & \vdots \\ \mathbf{V}_{m1} & \cdots & \mathbf{V}_{mk} \end{bmatrix}^{T}, \tag{12}$$

with equation (6), (8), we get

$$\mathbf{S} = \mathbf{A}\mathbf{Q}, \tag{13}$$

where $\mathbf{Q}$ is

$$\mathbf{Q} = \mathbf{Z}\mathbf{P}^{'}. \tag{14}$$

Algorithm 1 Off-line template matrix generation and PCA

Input:
all templates T_1 ...T_n
Output:
coefficient matrix $\mathbf{A}$, basis matrix Z
1: **for all** template T_i **do**
2: LoG (T_i) with CUDA parallel threads;
3: mean normalize T_i with CUDA parallel threads;
4: transform T_i to one column in the image matrix $\mathbf{T}_i^{'}$;
5: **end for**
6: $\mathbf{A}, \mathbf{Z} \Leftarrow PCA(\mathbf{T}_i^{'})$;

3.3. Elimination of unlikely image location

Rather than directly calculating equation (15), we speed up the calculation of $\mathbf{A}\mathbf{Q}$ by first eliminating image locations where a high score is unlikely. Specifically, we want to eliminate columns in $\mathbf{Q}$ and get $\mathbf{Q}^{'} = [\cdots \mathbf{Q}_i \cdots], i \in \mathcal{I}$, where $\mathcal{I}$ is a small subset of index into the image matrix columns corresponding to the probable image location. The final score matrix $\mathbf{S}^{'}$ will be

$$\mathbf{S}^{'} = \mathbf{A}\mathbf{Q}^{'}. \tag{15}$$

We use the L2 norm of each column in matrix $\mathbf{Q}$, i.e.,$\|\mathbf{Q}_j\|$, as a measurement. We prove that if $\|\mathbf{Q}_j\|$ is bounded above by λ, then every element $\mathbf{S}(i,j)$ in the corresponding column $\mathbf{S}(j)$ in the large Score matrix is bounded above by λ, that is, all templates score at this pixel is below λ. This observation allows us to eliminate unlikely image patches corresponding to j-th column in $\mathbf{Q}$.

Theorem 3.1 *if* $\|\mathbf{Q}_j\| < \lambda$*, then every element* $\mathbf{S}(i,j) < \lambda$.

Based on equation(15), $\|\mathbf{S}(j)\| = \|\mathbf{A}\mathbf{Q}_j\|$. Since A is an orthogonal matrix, $\|\mathbf{A}\mathbf{Q}_j\| = \|\mathbf{Q}_j\|$. Therefore, if $\|\mathbf{Q}_j\| < \lambda$, then $\|\mathbf{S}(j)\| < \lambda$. Therefore, $\sum_{i=1}^{m} \mathbf{S}(i,j)^2 < \lambda^2$. Therefore, each single value $\mathbf{S}(i,j) < \lambda$.

In experiments, if we choose the λ properly, the column number in $\mathbf{Z}\mathbf{P}'$ can decrease to a small number. The calculation time of $\mathbf{Z}\mathbf{P}'$ will decrease at least 50% using the PCA and candidate elimination method described above.

Algorithm 2 On-line massive template matching

Input:
template coefficient matrix $\mathbf{A}$, basis matrix $\mathbf{Z}$, threshold λ

Output:
Score matrix $\mathbf{S}$ for each frame

for each new frame I **do**
  $LoG(I)$ with CUDA parallel threads;
  call one parallel thread each image patch P_j;
  for all cuda threads **do**
    mean normalize P_j;
    transform P_j to one column in the image matrix $\mathbf{P}'_j$;
  end for
  compute $\mathbf{Q} \Leftarrow \mathbf{Z}\mathbf{P}'$ with CUDA parallel threads;
  $m \Leftarrow 1$;
  for $j = 1$ to N **do**
    if $\|\mathbf{Q}_j\| > \lambda$ **then**
      $\mathbf{Q}'_m \Leftarrow \mathbf{Q}_j$;
      $m \Leftarrow m + 1$;
    end if
  end for
  $\mathbf{S} \Leftarrow \mathbf{A}\mathbf{Q}'$
end for

4. Experiments

All experiments were performed using a standard desktop computer (Intel Xeon(R) CPU E5-2609 v2 @ 2.50GHz, 32G RAM) with an off-the-shelf GPU (NVIDIA Quadro K6000, 12GB GDDR5). We implement Vectorized NCC algorithm using NVIDIAs Compute Unified Device Architecture (CUDA) for real-time massive template matching.

4.1. Model and templates generation

For all 3D mesh models of the objects, they were generated by using Autodesk 123D Catch, the online free 3D modeling apps which generate 3D model from photos. We further use OpenGL rendering program to generate the object templates from different viewpoints.

4.2. Accuracy and speed evaluation

For each image sequence, we obtain the ground-truth rotation angles by manual annotation using EPNP [21]. The accuracy performance is quantitatively evaluated by the following equation:

$$err_i = |\theta_i - \alpha_i| \tag{16}$$

Where i represents the azimuth, elevation and yaw angle respectively, θ_i is the ground truth rotation vector, and α_i is the estimated rotation vector.

We evaluate our method on five objects with 218 images. Different from the feature-based approach, our method does the exhaustive search and can produce multiple hypothesis for each frame in the real-time system. Specifically, we divide the result matrix into twelve regions and find the maximal value and its position in each region. Table 1 shows the average error for each estimated angle of top one hypothesis, top three hypotheses and top ten hypotheses respectively.

Table 1. Average error of estimated angles

	yaw	elevation	azimuth
Top one hypothesis	2.4454	5.7676	17.4349
Top three hypotheses	1.347	3.2638	10.512
Top ten hypotheses	0.347	1.2638	2.512

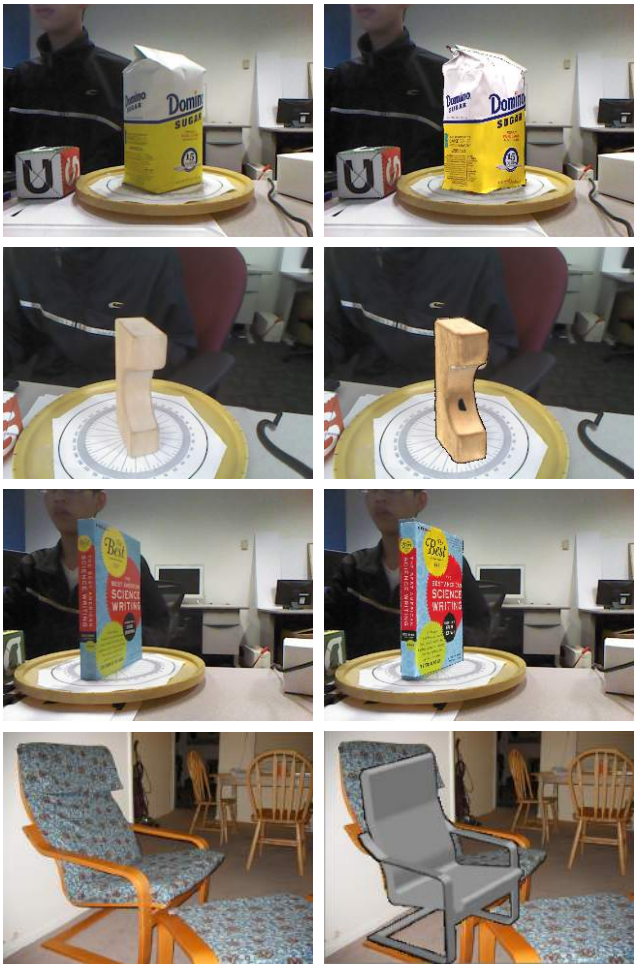
Table 2 shows the run-time comparison of OpenCV gpu-based NCC, our method of vectorized NCC, and the vectorized NCC with PCA and candidate elimination. The image size is 320x240, the template size is 180x180. Our method can process 4860 templates per second, that is, decreasing the processing time for each template to 0.206 ms. Our real-time detection-based pose estimation approach achieves the speed of 14 frames per second.

Table 2. Template number per second

OpenCV NCC	V-NCC	V-NCC with PCA
195	3980	4860

References

[1] Bay H, Ess A, Tuytelaars T, et al. Speeded-up robust features (SURF)[J]. Computer vision and image understanding, 2008.



(a) Frame (b) Aligned result

Figure 4. Example frames and pose estimation result. 3D model are aligned to the image to visualize the accuracy of our method.

[2] D. Lowe, Distinctive image features from scale-invariant keypoints, IJCV, 2004.

[3] Collet A, Berenson D, Srinivasa S S, et al. Object recognition and full pose registration from a single image for robotic manipulation[C] Robotics and Automation, ICRA, 2009.

[4] H. Barrow, J. Tenenbaum, R. Bolles, and H. Wolf, Parametric correspondence and chamfer matching: Two new techniques for image matching, in IJCAI, 1977.

[5] C. Olson and D. Huttenlocher, Automatic target recognition by matching oriented edge pixels, IEEE Transactions on Image Processing, 1997.

[6] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab, Dominant orientation templates for real-time detection of texture-less objects, in CVPR, 2010.

[7] D. M. Gavrila, A Bayesian, exemplar-based approach to hierarchical shape matching, PAMI, 2007.

[8] M. Ulrich, C. Wiedemann, and C. Steger, CAD-based recognition of 3d objects in monocular images, ICRA 2009.

[9] Marr D, Hildreth E. Theory of edge detection[J]. Proceedings of the Royal Society of London. Series B. Biological Sciences, 1980.

[10] Lewis J P. Fast normalized cross-correlation[C] Vision interface, 1995.

[11] Ino F, Gomita J, Kawasaki Y, et al. A GPGPU approach for accelerating 2-D/3-D rigid registration of medical images[M] Parallel and Distributed Processing and Applications, 2006.

[12] Babenko P, Shah M. MinGPU: a minimum GPU library for computer vision[J]. Journal of Real-Time Image Processing, 2008

[13] Gangodkar D, Gupta S, Gill G S, et al. Efficient variable size template matching using fast normalized cross correlation on multicore processors[M] Advanced Computing, Networking and Security, 2012.

[14] Zhao F, Huang Q, Gao W. Image matching by normalized cross-correlation[C]. Acoustics, Speech and Signal Processing, 2006.

[15] Di Stefano L, Mattoccia S. Fast template matching using bounded partial correlation[J]. Machine Vision and Applications, 2003.

[16] Di Stefano L, Mattoccia S, Mola M. An efficient algorithm for exhaustive template matching based on normalized cross correlation[C] Image Analysis and Processing, 2003.

[17] Di Stefano L, Mattoccia S, Tombari F. ZNCC-based template matching using bounded partial correlation[J]. Pattern recognition letters, 2005.

[18] Sarvaiya J N, Patnaik S, Bombaywala S. Image registration by template matching using normalized cross-correlation[C] Advances in Computing, Control, Telecommunication Technologies, ACT, 2009.

[19] Di Stefano L, Mattoccia S. A sufficient condition based on the Cauchy-Schwarz inequality for efficient template matching[C] Image Processing, 2003.

[20] Zhao F, Huang Q, Gao W. Image matching by multiscale oriented corner correlation[M] Computer Vision-ACCV 2006.

[21] Lepetit, Vincent, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. International journal of computer vision, 2009.

Shushman Choudhury and Mrinal Mohit

RISS 2014

Visual Pose Estimation for a Mobile Manipulator

Shushman Choudhury and Mrinal Mohit

Abstract—This report describes the contributions of the authors in the development of the vision pipeline for HERB - a bimanual robotic manipulator. We describe the architecture of the vision system, pose estimation using AprilTags and tracking of the same using Extended Kalman Filters. We also discuss our work towards developing ROCK (Robust Object Category and Kinematic Pose), a fast and efficient visual classification and pose estimation algorithm.

Keywords—Pose Estimation, AprilTags, Kalman Filtering, ROCK, HERB

1 Introduction

The objective of our work and research was to formulate and develop the vision system for HERB (Home Exploring Robotic Butler) [1] [2], a bimanual mobile manipulator. HERB is intended to perform useful tasks in general domestic environments, which lack the structure and predictability of factory environments. Accordingly, it needs a vision system that is comprehensive enough to identify objects of interest in clutter, is robust against failure, and is fairly efficient so as to provide actionable information to the other modules.

In this regard, our work addressed a number of issues of HERB's vision system. We developed a module for pose estimation and robust tracking of a visual fiducial system, using an Extended Kalman Filter model, and leveraging other insights about the pose observations we would obtain. We also made key contributions towards a novel algorithm for visual classification and pose estimation - particularly developing the database of object models, testing pose hypotheses, and obtaining the best result from multiple responses.

2 Overview of Hardware Platform

At the Personal Robotics Lab, a platform was needed that could operate synergistically with humans to perform tasks in the home. The design of HERB, therefore, reflects the research interest in human-aware two-arm manipulation of unstructured environments. HERB's hardware allows it to navigate indoors for hours at a time, sense its surroundings, and manipulate objects of interest for or with human partners, with minimal reliance on supporting infrastructure.

Fig. 1: The Hardware Platform - HERB

HERB's base comprises a Segway mobility platform, and it manipulates its environment with a pair of Barrett 7-DOF WAM arms with Barrett hands. Non-visual sensing is enabled by an array of four laser rangefinders just above ground level, odometric sensors in actuators and tactile sensors on the end-effectors. Computing is provided by 3 on-board high performance PCs while an ARM

• *The authors are with the Indian Institute of Technology Kharagpur, India. mrinal.mohit@iitkgp.ac.in, shushman.choudhury@cse.iitkgp.ernet.in*

microcontroller is used for low-level hardware control. The vision hardware is composed of a high-sensitivity monochrome camera by Allied Vision for texture data, and an Asus Xtion Pro Live for depth sensing.

The system can be remotely controlled over the network through ROS (Robot Operating System), and all modules for the robot (perception, planning, execution etc.) are based on the ROS nodes and topics architecture.

3 Vision Pipeline

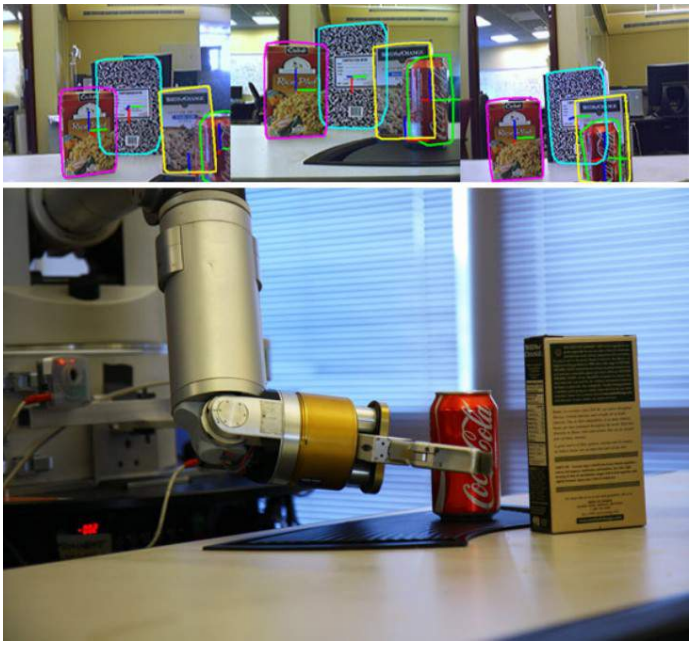


Fig. 2: Object detection for grasping - a use case

The primary requirement of HERB's vision system currently is to provide perceptual information to the planning and manipulation modules. Therefore, this information would be in the form of pose estimates of objects in HERB's current field of view, or which were seen recently.

HERB's image sensors provide the depth information of the immediate environment, as well as a high-definition video feed for obtaining color and texture information. We have envisaged HERB's vision system to comprise different modules that can run parallelly. Each of these modules obtains the image information from HERB's sensors and then applies its own algorithm to obtain relevant pose information. This information is then available on request to other modules, to use as they see fit. A schematic diagram is shown in Fig 3.

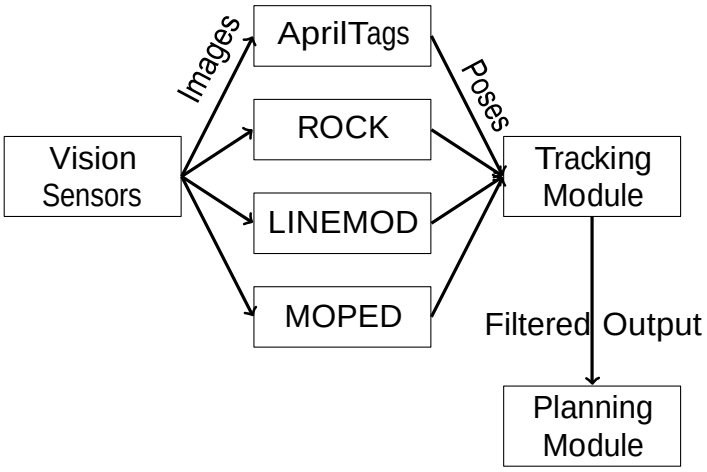


Fig. 3: The Vision Pipeline

Our work progressed in a number of phases, each of which is described in the subsequent sections.

4 Detection and Tracking of AprilTags

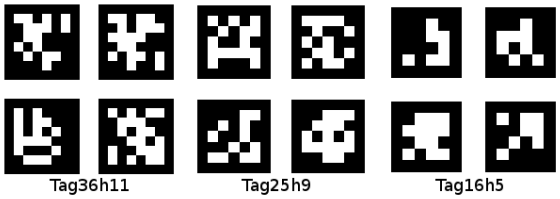


Fig. 4: Sample AprilTags

AprilTags [3] is a visual fiducial system that can be useful for a number of applications. The tags can be detected quickly and reliably through cameras, and for our purpose, the pose estimates of the surfaces on which tags are attached, can be obtained. They have the advantage of being robust to lighting and angle. The existing implementation of AprilTags was ported onto our system for use.

Furthermore, in view of the fact that HERB would often need information about moving objects, and thereby moving tags, and also to prevent the loss of tags for a few frames when occlusion occurs, a tracking module for the tags, with persistence against frame-loss, was developed. We used a simple Extended Kalman Filter (EKF) model [4] for this purpose.

4.1 EKF Model

Since the observations were of the pose estimate, but we were maintaining information that could predict its pose in the absence of observations, the measurement was of the form

$$z = [x\ y\ z\ q_x\ q_y\ q_z\ q_w]^T$$

i.e. the pose estimate of the tag in translation-quaternion format. Meanwhile, the state estimate of the tag was of the form

$$\hat{x} = [x\ y\ z\ q_x\ q_y\ q_z\ q_w\ \dot{x}\ \dot{y}\ \dot{z}\ \omega_x\ \omega_y\ \omega_z]$$

where the seven variables for the measurement are followed by the linear velocity ($\dot{x}$ etc.) and the angular velocity (ω_x etc.)

The model was based on the usual EKF formulation of a two-step cycle of prediction (before a measurement), and (subsequent) correction:

$$\hat{x_k}^- = f(\hat{x_{k-1}})$$

where f is the state transition model, $\hat{x_{k-1}}$ is the state after the previous time step, and $\hat{x_k}^-$ is the next predicted state. After the measurement z_k is obtained, the new state is obtained as follows:

$$\hat{x_k} = K_k z_k + (1 - K_k)\hat{x_k}^-$$

where K_k is a quantity called the Kalman Gain that is computed internally, based on the noise parameters of the model.

4.2 Tag Persistence

The tracking module was designed so as to persistently maintain the state of a tag that is being tracked when it is lost for a few observations. The predicted state of the tag is projected forward, as the error of the estimate magnifies. The number of frames for which persistence can last is a parameter that the user may set, after which the tracker for that tag becomes idle.

4.3 Pose Ambiguity Resolution

At certain perspectives, i.e. when the tag is not sufficiently skewed with respect to the camera, there are ambiguities in the solutions for the pose generated by the detection module. This results in two pose results being mathematically plausible, and the output of the module switching between the two, randomly, in successive frames.

This problem was tackled by leveraging two ideas - the first was to use a measurement validation gate to disallow the degradation of the state estimate from the Kalman Filter by an erroneous measurement. This requires the computation of a value called the normalized innovation squared,

$$c_v(k) = v(k)^T S(k)^{-1} v(k)$$

where $v(k)$ is the difference between the estimated state at the next time-step and the corresponding observation and $S(k)$ is a procedural matrix. If this value is out of (user-defined) bounds, then the measurement is rejected.

There was also the notion that ambiguous pose estimates would have similar re-projections in the camera frame, thereby providing a means to distinguish between a switched pose and the actual movement of the tags.

4.4 Preliminary Results

There was a reduction of the average error in detections and an improvement in the observed stability of the tag detections. The module was used as part of a demonstration for the movie "Robots 3D", a segment of which was shot by National Geographic in our lab. HERB's task was to clear up a dinner table by locating, grasping and manipulating them. Tags were fixed on objects to facilitate their localization in world space.

5 ROCK Detector

ROCK (Robust Object Category and Kinematic Pose) is a method for visual recognition and pose estimation of objects in a manner that is easily usable by HERB, and other platforms as well. A method is desired which is fairly lightweight, discriminative, tolerant to variance in viewpoint and illumination, can incorporate strong priors, and can have an adjustable trade-off between accuracy and complexity. As shown in Fig. 5, ROCK has a number of different modules, each with its own nuances and challenges. Our work was related to a certain portion of those modules.

5.1 Object Models

ROCK makes use of a database of object models, for recognition as well as pose estimation. The structure of these have been formulated for easy

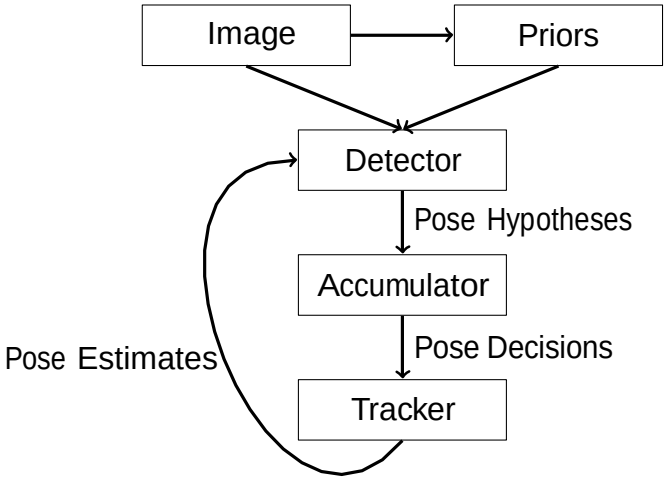


Fig. 5: ROCK Workflow

storage and processing of several dozen of them at a time, facilitating scalability.

We used Autodesk Inc's free software-service 123D Catch™ to generate fairly accurate and detailed three dimensional models from photographs of objects, shot under even lighting and fiducial backgrounds, from about two dozen perspectives (Fig 6). The software generates a mesh of vertices, edges and faces, which can be further refined and isolated from the background in 3D modelling packages. Furthermore, patches of the model, i.e small rectangular regions of interest, are processed and information about them stored, for use in the subsequent testing phase.

Fig. 6: 3D Mesh from 2D photographs

5.2 Pose Hypothesis Testing

Priors like geometric information of point features and other expectations about the object generate a number of pose hypotheses, each of which needs to be evaluated for likelihood. This is achieved by projecting the pattern for the pose onto the image and testing the expectation.

The idea behind this is that the difference between corresponding patches (in LAB color space) is fairly stable with respect to illumination and rotation. During the model creation phase, information about the patches of the model had been stored. For a patch P defined by a bounding box $[x1 < x < x2, y1 < y < y2]$, it is characterized by averaging the L,A,B channels values of the patch:

$$P(l) \equiv mean \left[\sum_{i=y_1}^{y_2} \sum_{j=x_1}^{x_2} L(i, j) \right]$$

$P(a)$ and $P(b)$ are calculated similarly, using the A and B channels respectively. Thereafter, a patch P is represented as:

$$P = \{P(l), P(a), P(b)\}$$

The difference between pairs of nearby patches on the model is stored as a binary vector

$$d(P_1, P_2) = \begin{bmatrix} P_1(l) > P_2(l) \\ P_1(a) > P_2(a) \\ P_1(b) > P_2(b) \end{bmatrix}$$

This is the vector that represents the relationship between a given pair of patches. The difference vectors for every such pair, is stored a priori for each model. For both training and test images, the patch information is computed quickly using the concept of integral images [10].

For each pose, the model is overlaid on the test image and the differences between patches are obtained and compared with the corresponding difference vectors of the model. The response (strength) of a pose hypothesis is therefore

$$R_H = \sum_i sim(M_{d_i}, H_{d_i})$$

where *sim* refers to a similarity metric (cosine similarity, table lookup), M_{d_i} refers to a difference vector for a pair of patches in the model, and H_{d_i} for the hypothesis.

A variety of similarity metrics are being explored for this purpose to account for noise, randomness and occlusion. For instance, a stricter

check mechanism reduces the false positives but also increases the effect of noise.

The basic idea is to start testing with coarse samples and then progressively refine the promising solutions. Testing at finer layers takes more time but gives more accurate responses, as one would expect, and this accuracy/computation tradeoff is an interesting area of exploration.

5.3 Pose Response Accumulation

The first stages of ROCK generate a number of object pose hypotheses with their respective responses, as mentioned earlier. From this information, a decision needs to be made about the number of instances of a particular object, and the most likely pose of each. For this purpose, we use mean-shift clustering techniques. [5] [6] [7]

For a set of pose hypotheses for a particular object, we first group them based on their position in 3D space, i.e. based on the translational component. This is done via mean-shift clustering with Euclidean distance. The bandwidth for this clustering is based on the known dimensions of the object, so as to disambiguate between hypotheses that must belong to different instances. Clustering in 3D space is a fairly fast procedure, and allows us to separate poses into groups of interest, by leveraging our prior knowledge about the object in question.

The next step is to analyze each of the small clusters obtained from the first step, and to identify if there is one or more instances of the object, and the best pose for each. For this, we use a non-linear mean-shift clustering for poses. The number of centres for each cluster indicates the number of instances that we believe are present, and the weighted average associated with each centre represents the accumulated pose hypothesis for that instance.

5.4 GPU Parallelization

The inherently independent nature of the patch comparisons and reprojections, for evaluation of the response of a test pose against an image for a given model, naturally incentivizes an exploration of parallel computing techniques to speeden up the process.

We made use of NVIDIA Corporation's CUDA™ , a parallel computing architecture and API, implemented on Graphics Processing Units (GPUs), rather than on traditional processors. Although the platform has proven to be highly successful in general, certain restrictions on the expected organization of the data structures in order to minimize bottlenecked memory transfer delays required re-organization of the ROCK Detector module.

5.5 Preliminary Results

We used a simple framework to test the strength of our preliminary detections. A number of pose hypotheses for an object were generated, with the assumption that it was on a table. This meant keeping the z coordinate and the angles about x and y axes constant and varying the other 3 components of the pose. Some of the poses were made to be near the ground truth. Having obtained the responses, we accumulated them as discussed and returned the best averaged pose. The initial results are encouraging, as is shown with one case (Fig 7). The black mesh is the visualization of the best averaged pose.

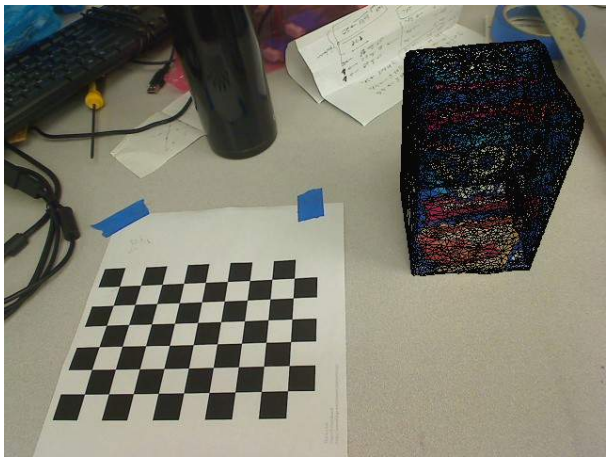


Fig. 7: The best response over several tested hypotheses is represented by the black mesh

6 LINEMOD Extensions

The recently proposed LINEMOD algorithm [8] has fared well in generic rigid object detection, proceeding using very fast template matching. It combines multiple complementary modes (e.g. images and depth maps), is fast, and can handle untextured objects.

The algorithm was tested on the same objects as were used in the aforementioned detectors. Experiments resulted in high true positives (˜85%) and low false positives (<10%) with high tolerance (˜50%) to occlusions. However, robust and effective detections require a high number of templates (2,000-10,000), leading to a time-consuming, tedious and unscaling model capture stage. We thus investigated a procedure for automated generation of these templates from dense meshes (already built as described in Section 5.1)

6.1 Automated Template Generation

We set up a 3D modeling and rendering package to capture color and depth images of objects so as to simulate the physical RGBD camera. For every object, a camera captures these frames from multiple viewpoints as observed from equally spaced points on different-sized spheres enclosing the object (Fig 8). These viewpoints approximate physically moving the camera around the object. We adapted LINEMOD to use these captured frames for model generation.

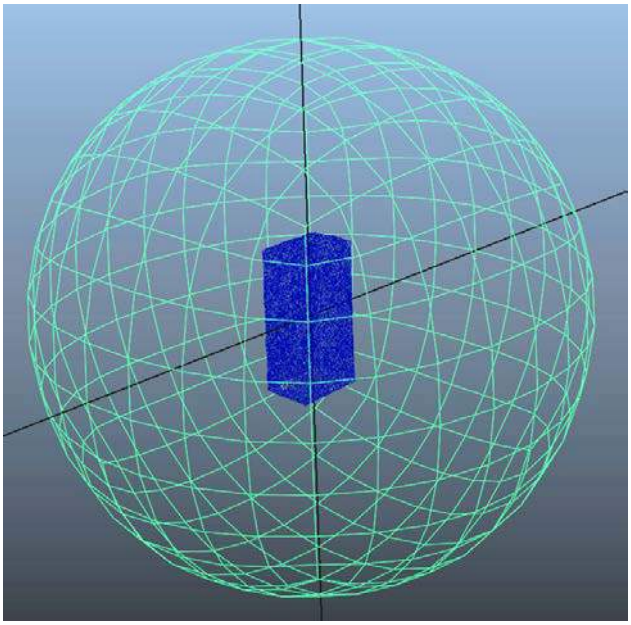


Fig. 8: Object images are captured from the points shown on the green dome.

Although the LINEMOD detection responses were expectedly good on synthetic testsets, the real-world performance turned out to be worse than that from a smaller number of hand-captured templates. The number of false positives were unacceptably high (>60%), and the algorithm slowed down by an order of magnitude. Possible reasons for this inadequacy were proposed - disparities between the actual and simulated camera frames, lack of fidelity in the generated meshes and idiosyncrasies in the template matching metric for simulated streams, but none of the hypthoses could satisfactorily explain the discrepancies.

A newer work on LINEMOD [9] embraces the automatic template generation concept, but has not been made publically available yet.

7 Miscellany

We also contributed to a number of other tasks, some of which are outlined here.

7.1 Camera Intrinsics Refinement

The intrinsic parameters of a camera are based on the pinhole model, and affect how the camera forms images from the environment.

Computationally, the intrinsic parameters are represented as a matrix, called the camera matrix:

$$C = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

where f_x and f_y are the focal lengths, and c_x and c_y refer to the co-ordinates of the image.

Accurate intrinsics are important for correct transformations from the 2D image space to the 3D world space. With a mixture of calibration methods and parameter tuning, we were able to enhance the performance of HERB's cameras.

7.2 Extrinsic Calibration of Vision Sensors

In order for HERB to interact accurately with objects in the real world, it is important for the relative positions of certain components of HERB, to be accurately known. The same applies to the vision sensors.

We wish to obtain T_{AC}, the transformation from the actuator to camera, which is fixed and unknown. Having visual samples of markers in the world from different camera positions, and knowing that they are fixed with respect to the world, we can formulate an expression for the extrinsic parameters:

$T_{WA_n} \equiv$ World to actuator (known)
$T_{WC_n} \equiv$ World to camera (unknown)
$T_{WC_n} = T_{WA_n} * T_{AC}$
$T_{CM_n} \equiv$ Camera to marker (known)
$T_{WM_n} \equiv$ World to marker (fixed; unknown)
$T_{WM_n} = T_{WC_n} * T_{CM_n}$

where n is the time. We will try to solve for T_{AC}, given several samples of T_{WA_n} and T_{CM_n}. Considering that the observed markers are fixed w.r.t the world, we obtain:

$$
\begin{aligned}
& T_{WM_1} = T_{WM_2} \\
\implies & T_{WC_1} * T_{CM_1} = T_{WC_2} * T_{CM_2} \\
\implies & T_{WC_2}^{-1} * T_{WC_1} = T_{CM_2} * T_{CM_1}^{-1} \\
\implies & (T_{WA_2} * T_{AC})^{-1} * (T_{WA_1} * T_{AC}) = K_2 \\
\implies & T_{AC}^{-1} * \ T_{WA2}^{-1} * T_{WA_1} \ * T_{AC} = K_2 \\
\implies & T_{AC}^{-1} * K_1 * T_{AC} = K_2 \\
\implies & K_1 * T_{AC} = T_{AC} * K2
\end{aligned}
$$

This is the well-known $AX = XB$ problem formulation that arises for the calibration of sensors, and we set this up as a system of equations by solving for the individual elements of T_{AC}.

7.3 Pipeline Optimization

Vision-based ROS nodes were modified such that they would be dormant unless subscribed to by an interested module. This lazy evaluation scheme led to savings in computation resources, and an increase in efficiency.

The detectors and the tracker were bundled into a common package, which launched automatically on robot start-up.

8 Conclusion

The work on AprilTags led to important insights about how to track these visual fiducial markers, and enhanced their usability. Furthermore, we were able to apply our knowledge and directly contribute towards a working demonstration. We complemented the implementation aspect of our work with a principled approach towards the new framework we are helping to develop for visual recognition and kinematic pose estimation, ROCK.

Finally, the auxiliary tasks we did by investigating existing systems such as LINEMOD, and working on the vision system components, will be important for other future applications.

Acknowledgments

The authors would like to thank Prof. Siddhartha Srinivasa, masters' student Aaron Walsman and the members of the Personal Robotics Lab for their guidance and advice, and the Robotics Institute Summer Scholar program for enabling this collaboration.

References

[1] S. Srinivasa, D. Berenson, M. Cakmak, A. Collet Romea, M. Dogar, A. Dragan, R. A. Knepper, T. D. Niemueller, K. Strabala, J. M. Vandeweghe, and J. Ziegler, "Herb 2.0: Lessons learned from developing a mobile manipulator for the home," *Proceedings of the IEEE*, vol. 100, no. 8, pp. 1–19, July 2012.

[2] S. Srinivasa, D. Ferguson , C. Helfrich, D. Berenson, A. Collet Romea, R. Diankov, G. Gallagher, G. Hollinger, J. Kuffner, and J. M. Vandeweghe, "Herb: a home exploring robotic butler," *Autonomous Robots*, vol. 28, no. 1, pp. 5–20, January 2010.

[3] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2011, pp. 3400–3407.

[4] R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," *Transactions of the ASME. Series D, Journal of Basic Engineering*, vol. 83, pp. 95–107, 1961.

[5] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Trans. Inf. Theor.*, vol. 21, no. 1, pp. 32–40, Sep. 2006. [Online].

[6] R. Subbarao, Y. Genc and P. Meer, "Nonlinear mean shift for robust pose estimation" *Applications of Computer Vision, 2007. WACV'07*, Feb. 2007

[7] D. Comaniciu and P Meer, "Mean shift: A robust approach toward feature space analysis." *IEEE Transactions on Pattern Analysis and Machine Intel ligence*, vol. 24, no. 5, pp. 603-619, 2002

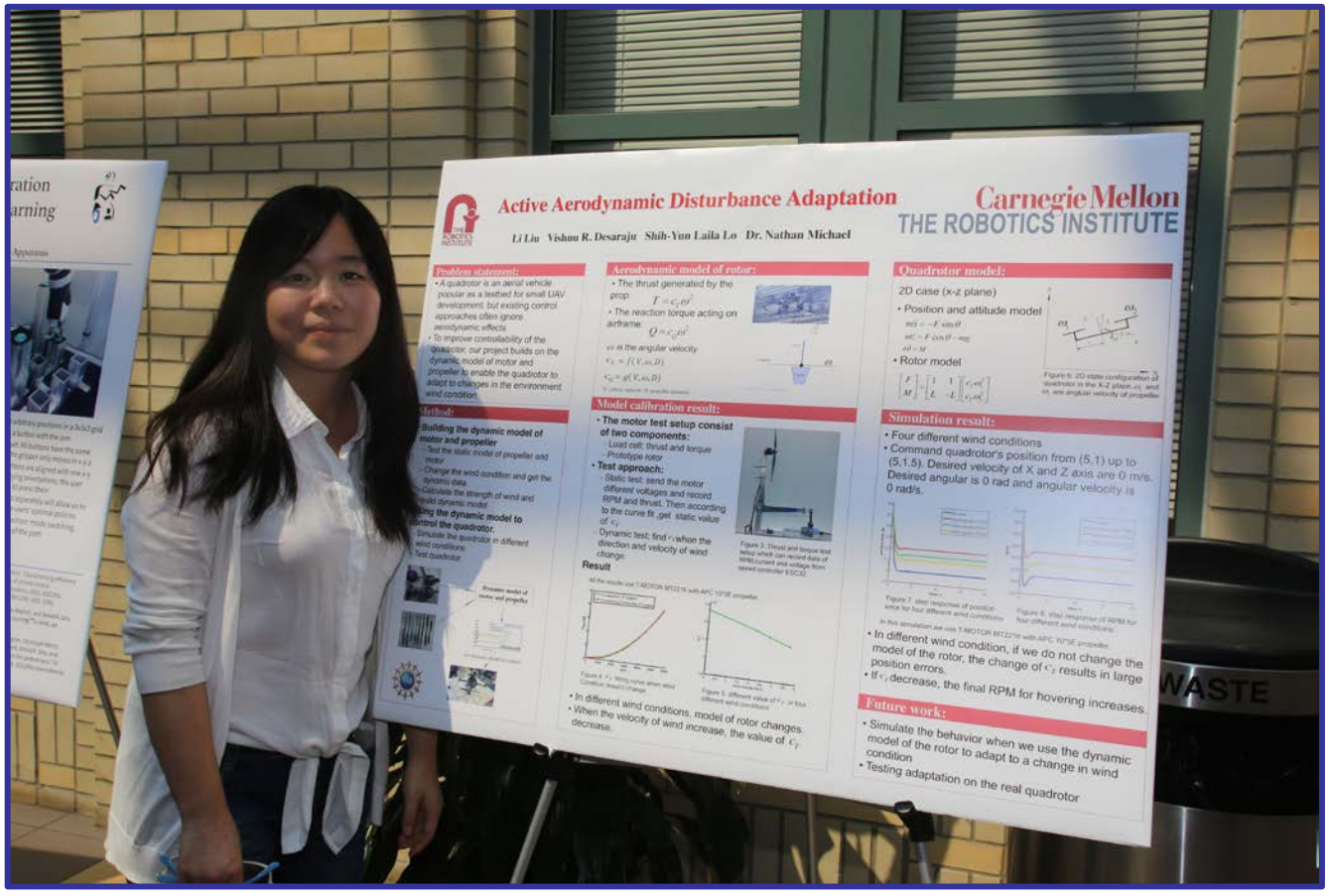
[8] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit, "Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes," 2011.

[9] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, , and N. Navab, "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," 2012.

[10] Viola, P., and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on* (Vol. 1, pp. I-511). IEEE.

Li Liu

RISS 2014



Active Control of Aerodynamic Disturbance Adaptation

Li Liu, Vishnu R. Desaraju, Shih-Yun Lo, and Nathan Michael

***Abstract*—The quadrotor helicopter is an aerial vehicle popular as a testbed for small unmanned aerial vehicle development. Existing control approaches often ignore the known aerodynamic effects on the quadrotor. The accepted quadrotor model is based on a thrust and torque model with constant coefficients calculated by static thrust tests. Such a model is a reasonable assumption in the environment where wind conditions do not change, but the static model is no longer valid when the vehicle undertakes aggressive manoeuvres when the wind conditions change. To address this problem and improve controllability of quadrotors, this paper builds a dynamic model of motors and propellers by using the setup we designed. According to the dynamic model of, this paper enables the quadrotor to adapt to changes in the environment wind conditions.**

Keywords—dynamic model, wind disturbance, simulation

I. Introduction

The quadrotor is an emerging rotorcraft concept for unmanned aerial vehicle(UAV). The vehicle consists of four individual rotors attached to a rigid cross airframe, as show in Figure 1.

Control of the quadrotor is achieved by differential control of the thrust generated by each rotor. A hierarchical approach is common for quadrotors. The lowest level is the control of the rotor rotational speed, directly with respect to the thrust of four rotors. The next level is in control of vehicle attitude, and the highest level is in control of position along a trajectory.

In this paper, we builds a dynamic model for more precise control of the low-level rotor control, the control of the rotor. The rotor controller gets the thrust and moments it needs from the position controller and attitude controller. According to the equation describing the relationship between thrust, moments and rotor speeds, calculate the voltage of motor. However, the equation should be revised when wind condition changes. This paper introduces a method to build a dynamic model of the rotor that is robust to changes in wind conditions.

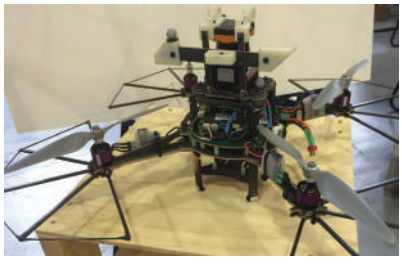


Fig.1 A quadrotor testbed with sensors

II. Aerodynamic Model of Rotor

A. Steady-state model of thrust and moment

The steady-state thrust generated by a rotor in free air can modeled as:

$$T_i = C_T \rho A_{r_i} r_i^2 \omega_i^2 \tag{1}$$

where, for rotor i, A_{r_i} is the rotor disk area, r_i is the radius, ω_i is the angular velocity of propeller, C_T is the thrust coefficient which depend on rotor geometry and profile, and ρ is the density of air. In practice, thrust can be models by a simple lumped parameter model:

$$T_i = c_T \omega_i^2 \tag{2}$$

Where c_T is often modeled as a constant.. In this paper,we tested c_T in static thrust test.

The reaction torque acting on the airframe generated by a hovering rotor in free air may be modeled as

$$Q_i = c_Q \omega_i^2 \tag{3}$$

where c_Q is the coefficient of the model of motors and propellers can be also impacted by rotor disk, radius and density of air and c_Q is also often used as a constant.

Our model assumes that quadrotor is hovering and the thrust is vertical to x-y plane. The total thrust at hover applied to the airframe is the sum of the thrusts from each individual rotor.

$$T_\Sigma = \sum_{i=1}^{4} |T_i| = c_T \left(\sum_{i=1}^{4} \omega_i^2 \right) \tag{4}$$

For quadrotors, we can write a model in matrix form to representing the relationship between thrust and torque to rotor speed

$$\begin{bmatrix} T_\Sigma \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} c_T & c_T & c_T & c_T \\ 0 & dc_T & 0 & -dc_T \\ -dc_T & 0 & dc_T & 0 \\ -c_Q & c_Q & -c_Q & c_Q \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \quad (5)$$

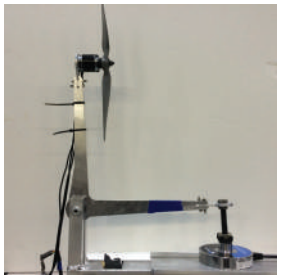
where τ_i (i =1, 2, 3) is net moment arising from the aerodynamics. Equation (5) shows that a change in wind conditions changes the model.

B. Static test results and c_T value in different wind conditions

Before we figure out the dynamic model of motors and propellers, we should test the static model which can be a basis of the dynamic model.

For the static thrust test, we built our motor test setup. The motor test setup we used consisted of two components, the load cell which we can use to get the value of thrust and torque and the prototype rotor, as shown in Figure 2a.

This setup also has a speed controller, which controls the rotor speed by adjusting the voltage and current of the motor. We can also get the data of rotor speed, the voltage and current of the motor from the speed controller.



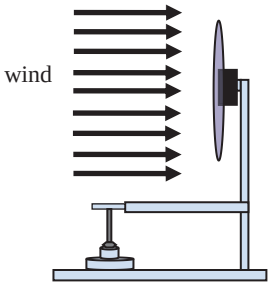


Fig.2 a) Thrust and torque test setup b) Dynamic testbed

To finish the static thrust test, we need to send different voltages to the motors and record the RPM of propeller and the thrust generated by rotor. Then, according to the curve fit, get static value of c_T , In this static thrust test, we used the T-MOTOR MT2216 with APC 10×5E propeller.

After running the static test, The static result is shown the value of c_T when wind conditions do not change as Figure 3. From the curve fit, c_T =1.266 $\times 10^{-7}$. And the value of c_T shows here is often used as a constant coefficient of the motor model.

The setup for the dynamic model is shown in Figure 2b. In this paper, we now only consider one direction of wind as shown in Figure 2b. In the dynamic test, we use a similar setup. Besides We use a wind velocity censer to get the different wind strength, and Figure 4 shows the value of c_T with changing wind velocity.

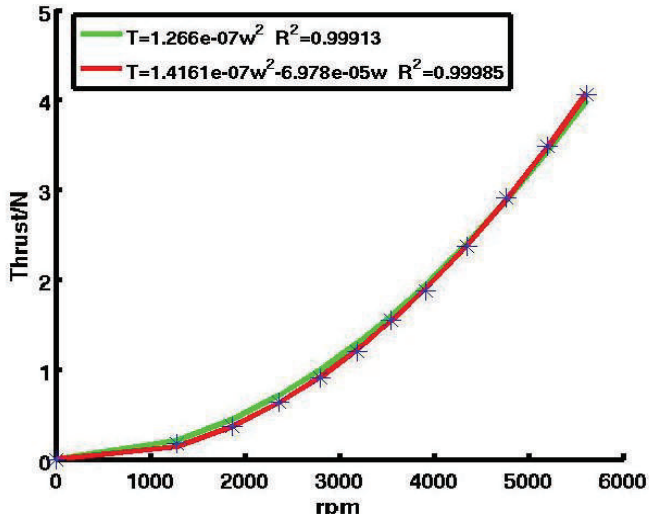


Fig.3 c_T fitting curve when wind condition doesn't change

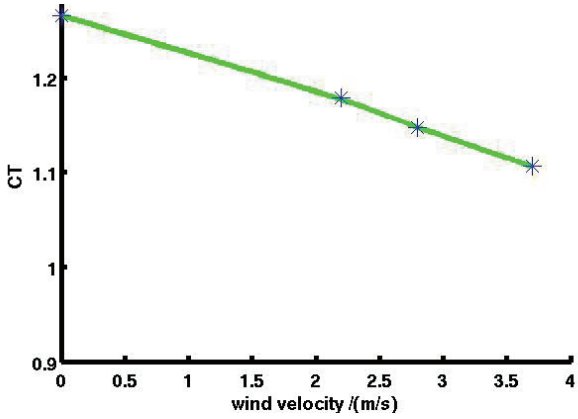


Fig.4 Different value of c_T in four different wind condition

III. 3D SIM.ULATION

A. 3D dynamic model

The quadrotor nested control method is shown as Figure 5. Most of the work in this paper is to figure out the values of u_1 and u_2 when wind condition changes. u_1 and u_2 are the voltage of motor.

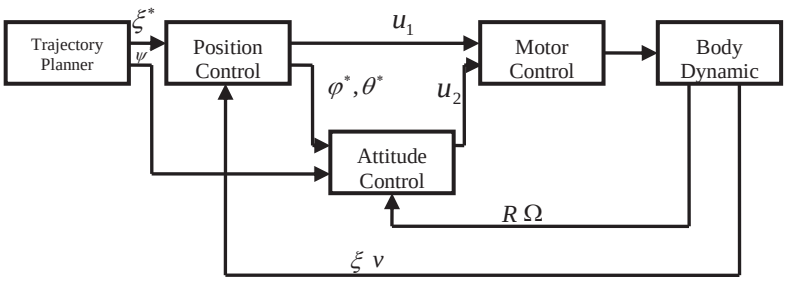


Fig.5 Nested feedback control frame of quadrotor

Let $\{A\}$ be the inertial frame, and $\{B\}$ be the body frame. The body frame is attached to the center of mass of the quadrotor, as shown in Figure 6. To get the rotation matrix R from body frame to inertial frame. We use the Euler angles to model the quadrotor's rotation in inertial first rotate about z_A by the yaw angle ψ, then about x_A by the roll angle ϕ, and finally about y_A by the pitch angle θ. The rotation matrix is

$$R = \begin{bmatrix} c\psi c\theta - s\phi s\psi s\theta & -c\phi s\psi & c\psi s\theta + c\theta s\phi s\psi \\ c\theta s\psi + c\psi s\phi s\theta & c\phi c\psi & s\psi s\theta - c\psi c\theta s\phi \\ -c\phi s\theta & s\phi & c\phi c\theta \end{bmatrix} \quad (6)$$

In this paper, cos is abbreviated to c and sin is abbreviated to s.

The motion equations of the airframe are

$$m\ddot{\mathbf{p}} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + R \begin{bmatrix} 0 \\ 0 \\ T_\Sigma \end{bmatrix} \quad (7)$$

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} d(T_2 - T_4) \\ d(T_3 - T_1) \\ M_1 - M_2 + M_3 - M_4 \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (8)$$

Where $\mathbf{p}$ is the position of quadrotor in inertial frame, m is the mass of the quadrotor, and p, q and r are the angular velocity of the quadrotor in body frame. The value of p, q and r are related to the derivation of the roll, pitch and yaw.

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} c\theta & 0 & -c\phi s\theta \\ 0 & 1 & s\phi \\ s\theta & 0 & c\phi c\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (9)$$

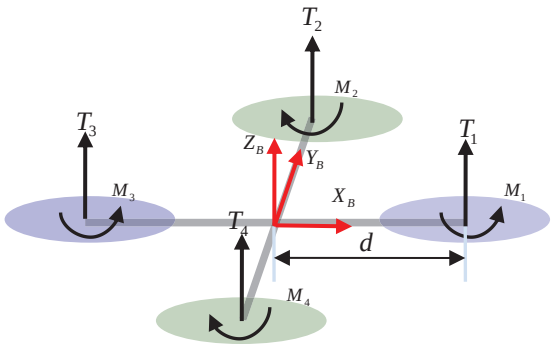


Fig. 6. Rigid body frame and thrust and torque acting on the quadrotor

According to the linearized model [2], we simplify the yaw angle to be a constant. The desired roll angle and pitch angle can be modeled by

$$\phi^* = \frac{1}{g}\left(\ddot{r}_1^d \sin\psi^* - \ddot{r}_2^d \cos\psi^*\right) \quad (10)$$

$$\theta^* = \frac{1}{g}\left(\ddot{r}_1^d \cos\psi^* + \ddot{r}_2^d \sin\psi^*\right) \quad (11)$$

The desired acceleration $\ddot{r}_i^d$ is from the feedback control. Here we use PD control

$$e_p = \mathbf{p}^* - \mathbf{p} \quad (12)$$

$$e_v = \dot{\mathbf{p}}^* - \dot{\mathbf{p}} \quad (13)$$

$$\ddot{r}_i^{des} = k_{p,i} e_{i,p} + k_{d,i} e_{i,v} + \ddot{r}_i^* \quad (14)$$

Where $k_{p,i}$ is the proportional coefficient of PD controller for rotor i, and $k_{d,i}$ is the derivation coefficient of PD controller for rotor i.

B. *Attitude simulation*

In this paper, we first give the result of attitude control. The attitude simulation based on the 3D dynamic model. We change the value of c_T in (5) to represent the wind conditions change. For the on-board attitude controller, we also use PD controller. Desired yaw angle ψ is set to zero. The initial status of roll, pitch and yaw are also set to zero. Desired roll angle is $\pi/10$, desired pitch angle is zero.

The error of roll angle from simulation is shown in Figure.7. Rise time for this step response is less than 0.5s. We want to make sure the rise time of attitude control is short enough to guarantee the convergence of the system. The units of angular error in figure 7 is meter and the time unit is second.

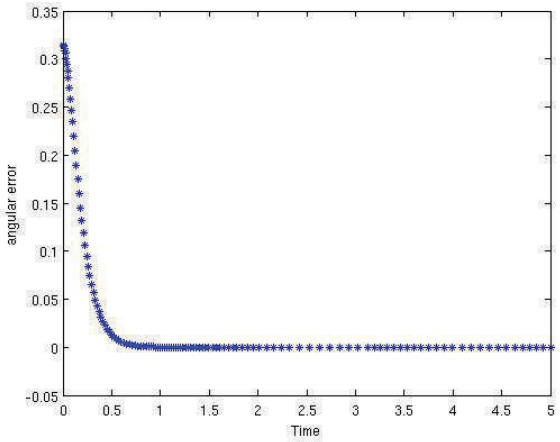


Fig. 7. Attitude error when close position control

C. Simulation result

These are the theoretical results we get for our control values in simulation. This simulation depends on model we describe in 3D dynamic model. The result here shows when wind conditions change and we do not change the model of motors and propellers, the control of the quadrotor will have offset.

The initial position of the quadrotor is (0,0,1) and desired position is (0.5,0.5,1.5). The Figure 8 shows the step response of the Z axis when wind conditions are constant.

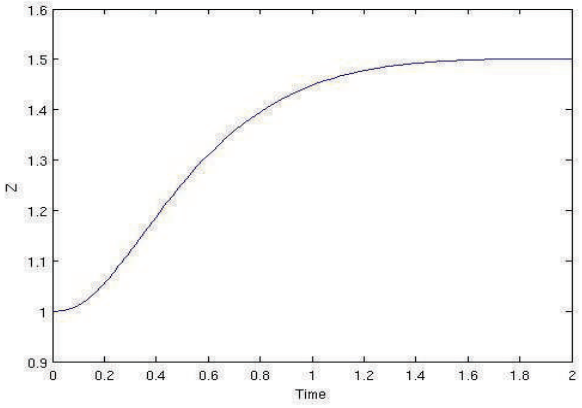


Fig. 8. Step response of Z axis when wind condition doesn't change

When we add the aerodynamic disturbance as shown in Figure 2b, the bigger the wind velocity the bigger the error in the end. The error of Z axis is shown in Figure.9. And the position changes of quadrotor is also shown in Figure.10.

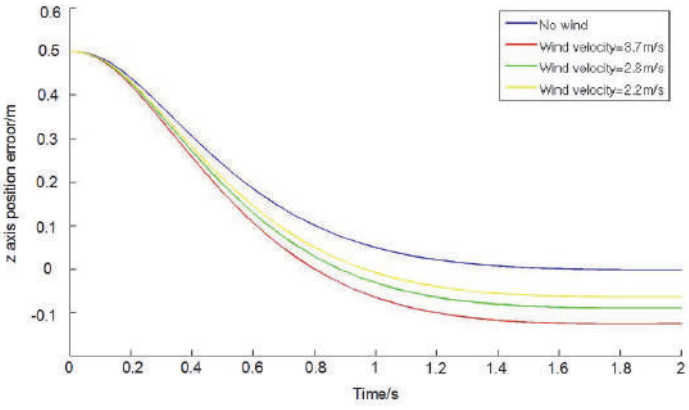


Fig. 9. Z axis step response error when wind condition changes

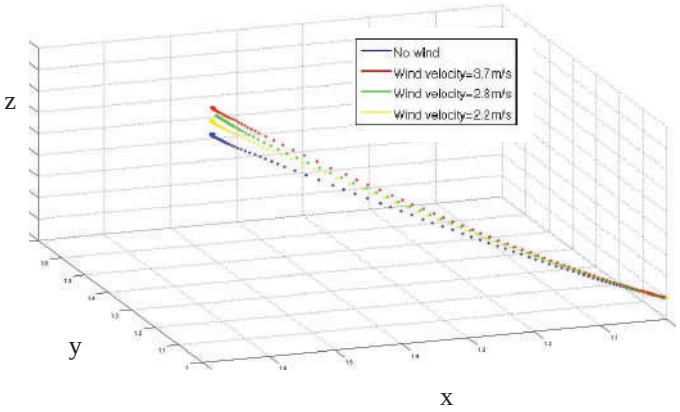


Fig. 10. Position of quadrotor in different wind conditions

These results show that the value of c_T changes with different wind velocities. That results in large position errors if we don't change our rotor model. The reason is that when wind conditions change, if we still use the static model, the motor will provide a constant speed and never decrease the offset.

Another observation is that if c_T decrease, the final RPM will increase. According to the equation (4), if motors want to provide the same thrust, the speeds of motors should increase.

Before figure out the specific relationship between wind conditions and c_T, we should also know the wind velocity without installing sensors because quadrotors should be lightweight. Current, voltage and RPM of rotor are provided by speed controller and we can plot the curve to find the equation between RPM and power. When wind conditions changes, the curve will change and this change is with respect to the value of c_T. In theory, the power of motor should be third power of RPM, but because of the aerodynamic effect the relationship changes.

The change of curve is shown in Figure 11. To make the figure clearly, the partial magnification of Figure 11 is shown in Figure 12. So when the quadrotor is flying, using the data of current, voltage and RPM of propellers we can sense the wind velocity.

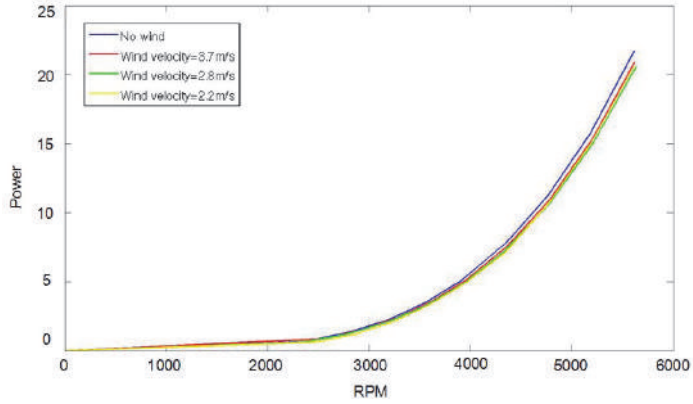


Fig. 11. Power and RPM changes when wind condition changes

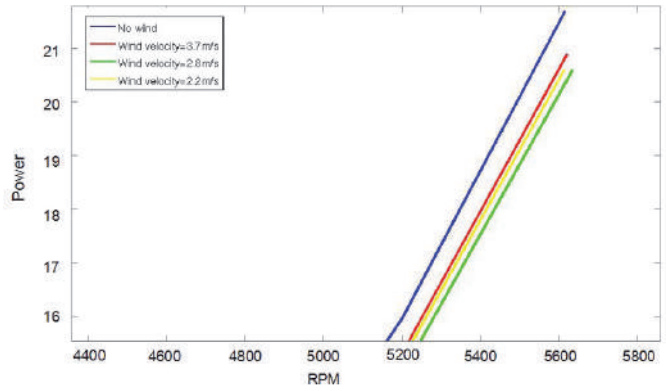


Fig. 12. Power and RPM changes partial magnification

IV. FUTURE WORK: CALCUATE C_T CHANGES

One future area of our research is to find the specific relationship between C_T and wind conditions. Now we now that when wind strength increase, the value of C_T will decrease.

We can learn the relationship between C_T and wind velocity by using machine learning, and after we know the wind velocity,we can change the model online.

V. CONCLUSION

This paper gives an approach to deal with the aerodynamic disturbance when quadrotor works, because when wind conditions are not constant, the model of rotor changes.

The static test gives the initial model of rotor and then we can test different C_T value when wind condition changes and find the relationship between C_T and wind conditions.

Finally, we need to be able to know the wind conditions without using sensors. Depending on the change of equation between power and RPM of propeller when wind condition changes, we can calculate the wind condition.

ACKNOWLEDGMENT

I would like to thanks to my mentor Dr. Nathan Michael, Vishnu Desaraju, Shih-yu Lo for the supervisor, Rachel Burcin, Keyla Cook and all the RISS coordinators for their help.

REFERENCES

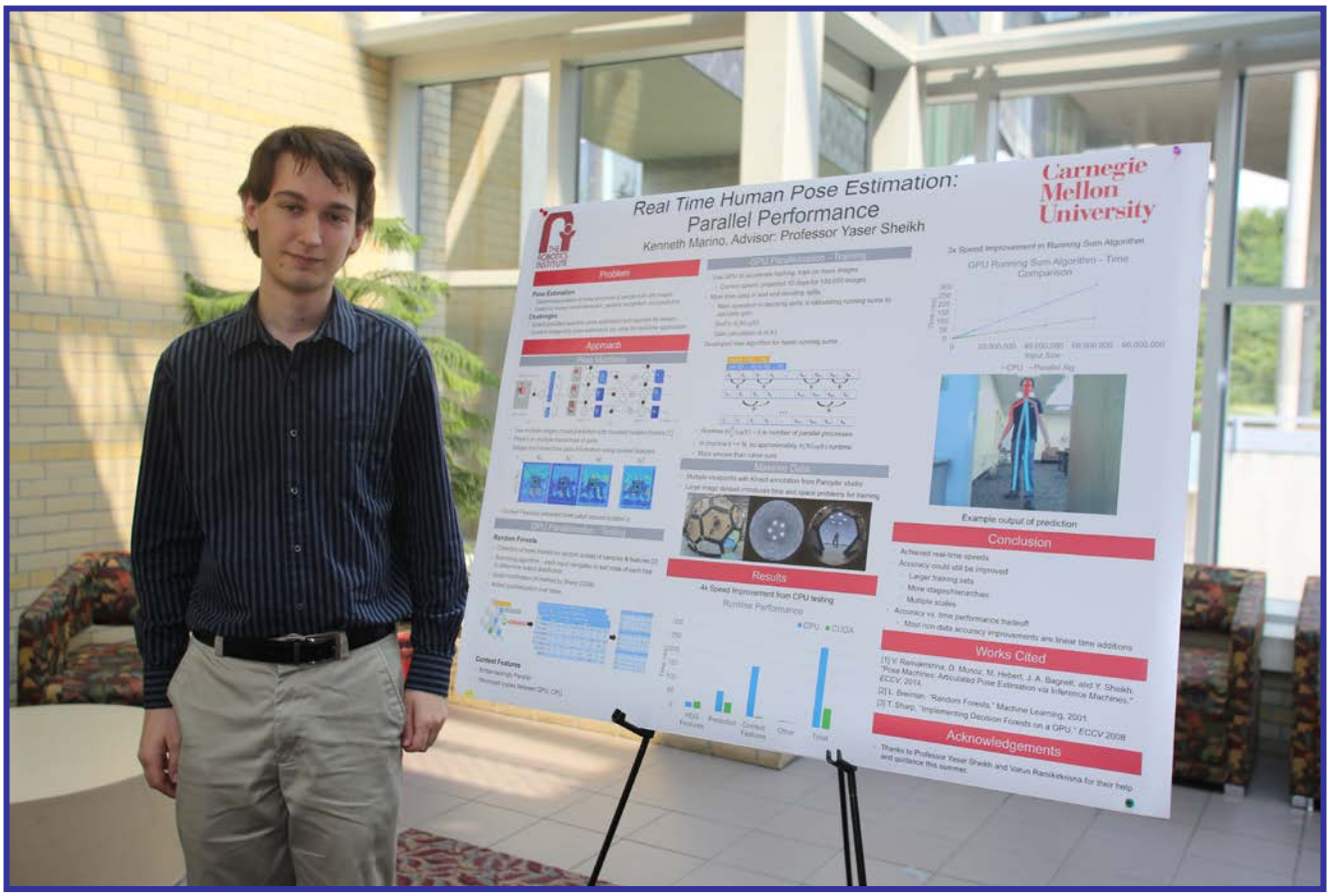
[1] Robert Mahony, Vijay Kumar and Peter Corke, "Multirotor Aerial Vehicles: Modeling, estimation, and control of quadrotor", IEEE Robotics & Automation Magazine 19, pp.20-32.

[2] Nathan Michael, Daniel Mellinger, Quentin Lindsey and Vijay Kumar, "The Grasp Multiple Micro-UAV Test Bed", IEEE Robotics and Automation Magazine 17, pp.56-65

[3] Gariel M. Hoffmann, Haomiao Huang, Steven L. Waslander and Claire J. Tomlin, "Quadrotor Helicopter Flight Dynamics and Control: Theory and experiment", AIAA Guidance, Navigation and Control Conference and Exhibit, 20-23 August 2007,Hilton Head, South Carolina.

Kenneth Marino

RISS 2014



Real Time Human Pose Estimation for Boosted Random Forests and Pose Machines

Kenneth Marino, Georgia Institute of Technology

***Abstract*— Current state-of-the-art for human pose estimation is in that it is unsuited for real-time performance without the addition of depth information, which can be a major limitation. In this paper, we extend the work on Pose Machines using GPU acceleration to achieve performance in real time. We also examine and propose solutions to the memory and time issues related to the training of Pose Machines with large datasets. These include a GPU accelerated algorithm for training Pose Machines and changing the way data is used, separating training data into "structure points" and "evaluation points." Finally, we examine the effect of these changes to the speed of testing and training.**

***Index Terms*—computer vision, inference machines, machine learning, pose estimation.**

I. Introduction

A. Problem

The essential problem of human pose estimation from still images can be described simply as the identification of the locations of a number of joints on a two dimensional image. For instance, in our system, we predict the location of the forehead, the base of the neck, the left and right shoulders, the left and right elbows, the left and right wrists, the left and right hips, the left and right knees, and the left and right ankles. An example pose can be seen in Figure 1.

We further specify that we do not have access to depth information. The Microsoft Kinect, which is currently the most robust real-time human pose estimator, collects a depth image using an infrared sensor, as well as an RGB image to do pose estimation [1]. While this allows for efficient prediction of human poses, it imposes a limitation in hardware. Most significantly, it makes pose estimation impossible for outdoor scenes where infrared sensing is ineffective. It also requires the use of specialized hardware which can limit its potential uses.

Another challenge of the problem is to do this prediction in real time. For example, we might like to take a live video stream and overlay part locations so that we can have a real time 2D pose for human subjects. This requires that prediction take place relatively quickly. The Kinect runs at approximately 30 frames-per-second, meaning that part prediction takes no more than about 33.3 milliseconds. Most current approaches to the problem with still images use a graphical model to capture relationships between parts. However, many of these approaches suffer from either accuracy problems because the models are too simple, or suffer from tractability problems because the model is too complex [2]. Clearly time complexity is an issue because inference on these models takes orders of magnitudes longer than what is required for real-time performance.

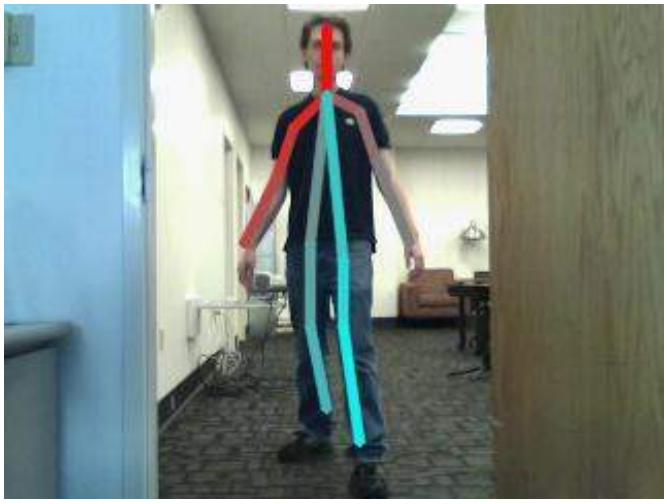


Fig. 1. An example generated pose. The colored line segments show the connection between the 14 annotated locations. The output pose gives the location of joints from which limbs can be inferred.

B. Previous Work

Until quite recently, most work on pose estimation from single images was based on using graphical models to capture dependencies between parts for prediction [3, 4, 5, 6, 7, 8], typically with simple tree or star-structured models. The problem, however, with simplified graphical models is that they do not capture a number of important dependencies, such as symmetric parts (to avoid double counting) and these methods often fail when certain parts are occluded in the image. Inference on exact graphical models is too difficult and computationally expensive, except for very simple models.

More recent work has examined using Deep Neural Networks to train and refine joint predictors [9]. Its use of refining predictions is similar to Pose Machines.

C. Pose Machines

Our approach is to extend the work of Ramakrishna et al on Pose Machines. Pose Machines are currently the state-of-the-art for single-image human pose estimation. This approach sidesteps the issue of representation by approaching pose estimation as a structured prediction problem. The prediction

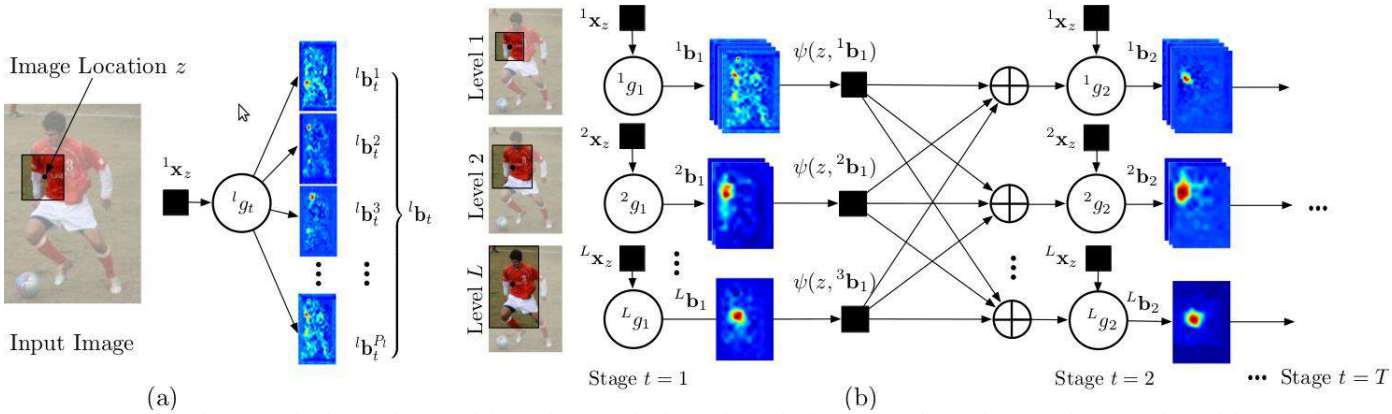


Fig. 2. Pose Machine from Ramakrishna et al. For each image location, the figure shows the layers of prediction from Level 1 to Level L and the multiple stages of prediction from 1 to T. Photo courtesy of Ramakrishna.

task is given an input image, we would like to predict the anatomical landmarks $Y = (Y_1, \dots, Y_P)$ for each of the P parts, in this case P=14 [2]. We predict each part such that $Y_p \in Z \subset \mathbb{R}^2$, where Z is the set of all pixel locations (u, v) in the image. The inference machine then consists of a sequence of multi-class classifiers, $g_t(.)$, which are trained to predict the location of each part. In each stage $t \in \{1, \dots, T\}$, we predict a confidence for each output assignment $Y_P = z, \forall z \in Z$ based on the input image data $x_z \in \mathbb{R}^d$ and contextual information from the previous stage, $\psi(z, b_{t-1}^i)$, where b_t^p represents the confidence at stage t that $Y_p = z, \forall z \in Z$. We can then compute the confidence that a particular part p belongs at location z by $b_t(Y_p = z) = g_t^p(x_z; \cup_{i=1}^P \psi(z, b_{t-1}^i))$, where the union operator is the concatenation of the outputs of $\psi(.)$ for each part.

The intuition behind this framework is that passing information between predictors for each stage allows for information about the location of different parts to be used in the next stage to predict other parts. For instance, a high likelihood of the head in one location might make it far more likely that the neck is at a location near it. This framework, we believe, implicitly captures the statistical relationship between these parts.

The predictors themselves are implemented as gradient boosted random forests classifiers. The basis of this algorithm is the random forest [10]. Given an input $x \in \mathbb{R}^d$, the algorithm returns the corresponding continuous output $y \in \mathbb{R}^P$. This output is determined in the following way. For each tree in the forest, the algorithm determines the leaf node that corresponds to a particular x_i. Each internal node keeps track of a dimension of x dim and a threshold $thresh$. The value of x_i is compared to the threshold at that dimension to determine if the corresponding leaf node is a left or right descendant of the current node – if $x_i[dim] \leq thresh$, it updates the current node to be the left child. Otherwise, it updates it to be the right child. Once it has reached a leaf node of the tree tr, it outputs the distribution of y that is stored there $P_{tr}(x_i)$. This distribution corresponds to the distribution of training examples that also navigated to this node during training. Once this is done for all trees, the final output y is simply the average of all of the $P_{tr}(x_i)$'s.

D. *Fundamental Time, Energy Tradeoff*

One of the fundamental trade-offs in many problems is between speed and accuracy. As one might expect, methods of pose estimation that are more accurate often take longer to compute. This is no different for Pose Machines. Below is a chart showing the major parameters of the Pose Machine Algorithm, how the value of that parameter effects the runtime of the algorithm and the value used subsequently to compare running times for this paper. Increasing each of these parameters will improve the accuracy of the pose estimator to some extent, but causes the prediction to take longer [2].

TABLE I
TUNABLE PARAMETERS

Parameter	Time Relationship	Value in System
Stages	Linear	3
Levels of Hierarchy	Linear	3
Depth	Sublinear	12
Number of Trees	Linear	20
Boosting Iterations	Linear	25
Image Resolution	Quadratic	360p

II. APPROACH

A. *CUDA Accelerated Run Time Performance*

Before work was started, running Pose Machines for a complete image took approximately 270 milliseconds using the parameters from Table I. This would mean that you could run at approximately 4 frames-per-second on a top-of-the-line consumer processor. Perhaps additional improvements in processors could bring this up to a reasonable frame-rate, but for now it is simply too slow to allow for real time applications.

A major focus of this work, then, is to increase the speed of this prediction without sacrificing accuracy. To that end, Pose Machines was accelerated using GPU acceleration.

The original work on GPU accelerated algorithms for random forests was performed by Toby Sharp at Microsoft Research. The basic concept of the algorithm was to take advantage of the parallel capacity available with graphics processors and compute the output of all of the inputs x_i, at the same time. Thus instead of computing the output $y_i = f(x_i)\ \forall i$ serially, they are computed in parallel. This is particularly

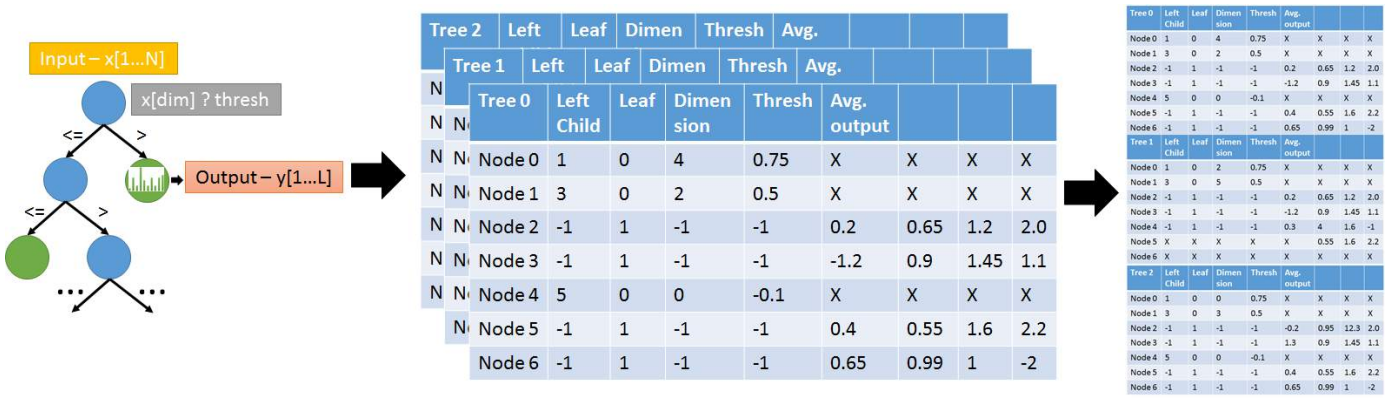


Fig. 4. GPU-optimized data structure for random forests. Sharp Trees are concatenated on top of each other to allow for additional parallel operations. Operations on the data structure are parallel over examples, as well as over trees.

useful with respect to pose machines because we are essentially computing this value for Z, the set of all pixel locations (u, v) in the image. Sharp's algorithm stores each tree as a matrix where each row of the matrix is a node in the tree. A modified version of this matrix is shown in Figure 3.

As before, each internal node stores a dimension and a threshold. Instead of doing a conditional branch to determine the successor node, the modified Sharp algorithm calculates the same comparison $x_i[dim] > thresh$ [11]. Now if the condition was true, the value in memory is 1, otherwise the value is 0. The algorithm then loads the index of the next node by calculating $nextnode = leftchild + x_i[dim] > thresh$. The way the tree is laid out, the left and right children of a node are always next to each other. The significance of doing the calculation this way is that no branching is ever performed. This means that all of these calculations can be done simultaneously for $\forall i$ without forcing the GPU to spin some of its threads, which is what would happen with a $if, else$ construct. Once all of the x_i's reach leaf nodes, the average outputs for all of the trees are averaged. Additionally, since we are in fact running boosted random forests, this is repeated for each boosting iteration and the final output is determined by the normal weighted sum of the outputs for each stage.

One additional change that was made from the Sharp Algorithm (besides the generalization of the conditions) is that the operation is further parallelized over trees (see Figure 4). In addition, while Sharp stores the node information (left child, leaf, dimension, thresh) in the same data structure as the average outputs, our modification puts them in separate data structures. These changes allow for more efficient GPU memory use and more efficient use of available threads.

In addition to the random forest prediction being performed on GPU, most of the operations of Pose Machines was moved onto GPU. The most significant calculation moved to GPU was the calculation of the context features, earlier simplified $\psi(z, b_{t-1}^i)$. Additional details of context features can be found in [11], but the basic idea is that the outputs from the previous stage for all of the parts are condensed into score maps and fed as input into the next prediction stage. In this case, each of these context features for each input x_i is completely independent, so this was fairly easily parallelized.

The boosted adding of the random forests was moved onto GPU. With all of the major calculations now done on GPU, memory copies between CPU and GPU (one of the most computationally expensive operations) were minimized.

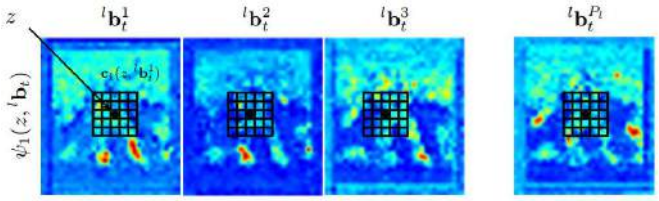


Fig. 5 Context Features Visualization. The output values for the locations surrounding each point are concatenated into a new input feature for the next stage of computation. Photo courtesy of Ramakrishna.

B. *GPU-Accelerated Training*

Runtime performance is an important part of real time human pose estimators, but the other half of the problem, besides the runtime performance, is the training performance. Currently, for a dataset size of one thousand, it takes about 1 day. Unfortunately, to achieve performance close to Kinect accuracy, the algorithm likely needs to train more on the order of 10 million images. Assuming an approximately linear runtime for training (this is likely optimistic). It would take approximately 10,000 days or about 27 years to train a model with 10 million images. Clearly this is intractable.

One approach to fixing this problem is to parallelize using GPU acceleration. For this, first we look at the basic algorithm for training a random forest (the backbone of the prediction, and the part that takes up most of the training time). The recursive algorithm for building a tree, along with the runtime with respect to the number of examples, of different operations is shown in Figure 6.

For parallelization, clearly the most important things to parallelize are the sorting of X, and the determination of the best gain on line 7. There are numerous parallel algorithms for sort, so we will concentrate on the determination of gain.

The determination of the gain basically boils down to the simple operation of determining running sums. The formula we use to calculate the gain for a regression tree is $err_{parent} - (err_{left} + err_{right})$. To calculate a particular error of a particular input dimension and split, the formula is $err = \sum_{j=1}^{D}(\sum_{i=1}^{N} w_i y_{ij}^2) - \frac{(\sum_{i=1}^{N} w_i y_{ij})^2}{\sum_{i=1}^{N} w\_i}$, where N is the number of samples and D is the dimensionality of the output. From this,

Algorithm 1: growTree

Input: $X, Y, active\_indices, active\_features$

1 **Compute weighted sum of** Y **and weighted sum of** Y^2;
2 **Compute parent node info;**
3 **Shuffle order of features to iterate over;**
4 **for** *feat in active_features* **do**
5 $SortX[idx][feat], idxpairsbyvalueofX$;
6 **for** *idx in active indices* **do**
7 $ComputeGain$;
8 $Keepifbest$;
9 **for** *idx in active_indices* **do**
10 **if** $X[idx][bestFeat] \leq thresh$ **then**
11 $active\_indices\_left.append(idx)$
 else
12 $active\_indices\_right.append(idx)$
13 **if** *not leaf condition* **then**
14 $growTree(X, Y, active_indices_left, active_features)$
15 $growTree(X, Y, active_indices_right, active_features)$

Fig. 6 Algorithm for growing a decision tree. Notice the bulk of the time is spent in the for loop starting on line 4. The running time of this part is $\theta(NlogN)$.

Algorithm 2: RunningSum

Input: $X[1...N]$

1 $height = \lceil \log_2 N \rceil$;
2 $chunkSize = 1$;
3 **for** *idx in active_indices* **do**
4 **Parallel: for** *each set of two chunks C1[1...chunkSize], C2[1...chunkSize]* **do**
5 **Parallel: for** $i = 1$ *to chunkSize* **do**
6 $C2[i] = C1[chunkSize]$;

Fig. 7 New Algorithm to compute running sums in parallel.

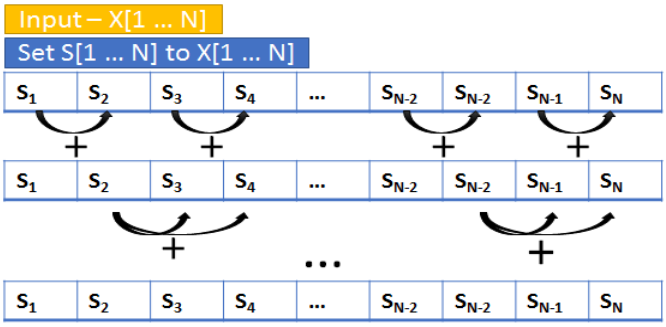


Fig. 8 Visualization of running sum algorithm in Figure 7.

you can see the bulk of the work is computing these sums. Thus, we developed the following algorithm for computing the running sum in parallel shown in Figure 7 and Figure 8.

The running time of this algorithm is $\theta(\frac{N}{k}logN)$, where k is the number of parallel processes available. The only serial work is related to the log of the size of the input. The adding operations occur in parallel, but because hardware has a fixed limit to the number of parallel operations, the actual time depends on that hardware constant k.

III. Results

Figure 9 shows the runtime performance of Pose Machines using the parameters from Figure 3 using naïve CPU parallelization and the GPU acceleration version.

These results show a great improvement in runtime from about 4 frames per second to about 17, and approximately 400% increase. In particular, the context feature calculation is greatly speeded by GPU acceleration, however, the prediction is still greatly sped up. The only operation not implemented on GPU was the Histogram of Gradients (HOG) feature computation. Clearly speeding up this computation could have a positive effect on runtime performance.

Figure 10 shows the speed results for the GPU accelerated running sum algorithm. For sufficiently large inputs, the GPU accelerated algorithm runs about 3 times the speed. While this result is certainly promising, GPU acceleration of training may not be the ideal solution. First, there is a memory issue in that even high-end GPU processors have only about 10 GB of memory which may prevent GPU from being used for training. In addition, given that GPU memory is filled, only one CPU process can be used at a time during training which prevents CPU acceleration of training. This means that unless

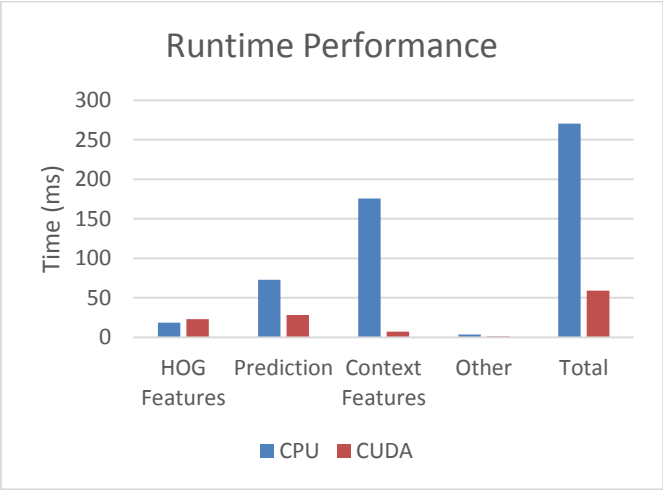


Fig. 9 Bar graph showing runtime performance of Pose Machines. All parameters are listed in Table I.

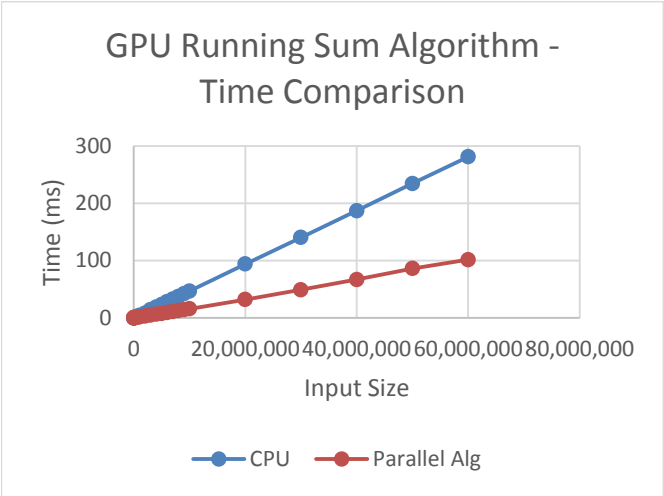


Fig. 10 Line graph showing performance of new running sum algorithm compared to simply calculating a running sum on CPU.

the GPU accelerated training is far faster than a single-core CPU algorithm, GPU accelerated training is unlikely to help.

IV. Conclusion

The most impressive result of the work is the time performance. Based on the number reported, running Pose Machines using GPU acceleration achieves performance close to that of the Kinect with respect to speed. It is currently the fastest implementation of a working pose estimator that does not use depth information. DeepPose [9] claims a running time of 100 milliseconds, compared to our 60 milliseconds.

The accuracy of the model is still the biggest problem. Pose Machines still do not match the prediction accuracy of the Kinect. One of the major problems right now is the limited training sets that the algorithm is trained on. Improvements in performance from GPU accelerated training and other improvements in training time such as the structure and evaluation point schema will be important to train the algorithm on extremely large datasets.

V. Future Work

Most of the future work on this project will concentrate on training larger datasets. This will involve solving problems surrounding training speed as well as problems with memory. Currently, the entire training dataset is put into memory for training, but as the dataset gets larger, this will become intractable.

Additional work is currently being done to train by using some input examples as "structure points" and used to determine splits and using the rest to determine the distribution on leaf nodes. Ideally this would reduce the runtime by reducing the number of "splits" that internal nodes have to consider during training. This work is ongoing, so no results have been generated, but the preliminary findings make this a promising area.

A possibility beyond directly solving the training time and space issues is to better use available training examples. The idea is to essentially learn which images will best improve the performance of the algorithm. The algorithm will train on one subset of the data and then predict the output for the other subset of the data. The images that the algorithm did poorly on will then be used in the next round of training so that it can better learn the things it missed.

References

[1] J. Shotton et al, "Efficient Human Pose Estimation from Single Depth Images," *CVPR*, 2011.

[2] V. Ramakrishna, D. Munoz, M. Hebert, J. A. Bagnell, and Y. Sheikh, "Pose Machines: Articulated Pose Estimation via Inference Machines," *ECCV*, 2014.

[3] P. F. Felzenszwalb, D.P. Huttenlocher, "Pictorial structures for object recognition," *IJCV*, 2005.

[4] D. Ramanan, D.A. Forsyth, A. Zisserman, "Strike a Pose: Tracking people by finding stylized poses," *CVPR*, 2005.

[5] M. Andriluka, S. Roth, B Schiele, "Monocular 3D Pose Estimation and Tracking by Detection," *CVPR*, 2010.

[6] M. Andriluka, S. Roth, B Schiele, "Pictoral Structures Revisited: People Detection and Articulated Pose Estimation," *CVPR*, 2009.

[7] Y. Yang, D. Ramanan, "Articulated pose estimation with flexible mixture-of-parts," *CVPR*, 2011.

[8] S. Johnson, M. Everingham, "Clustered pose and nonlinear appearance models for human pose estimation." *BMVC*, 2010.

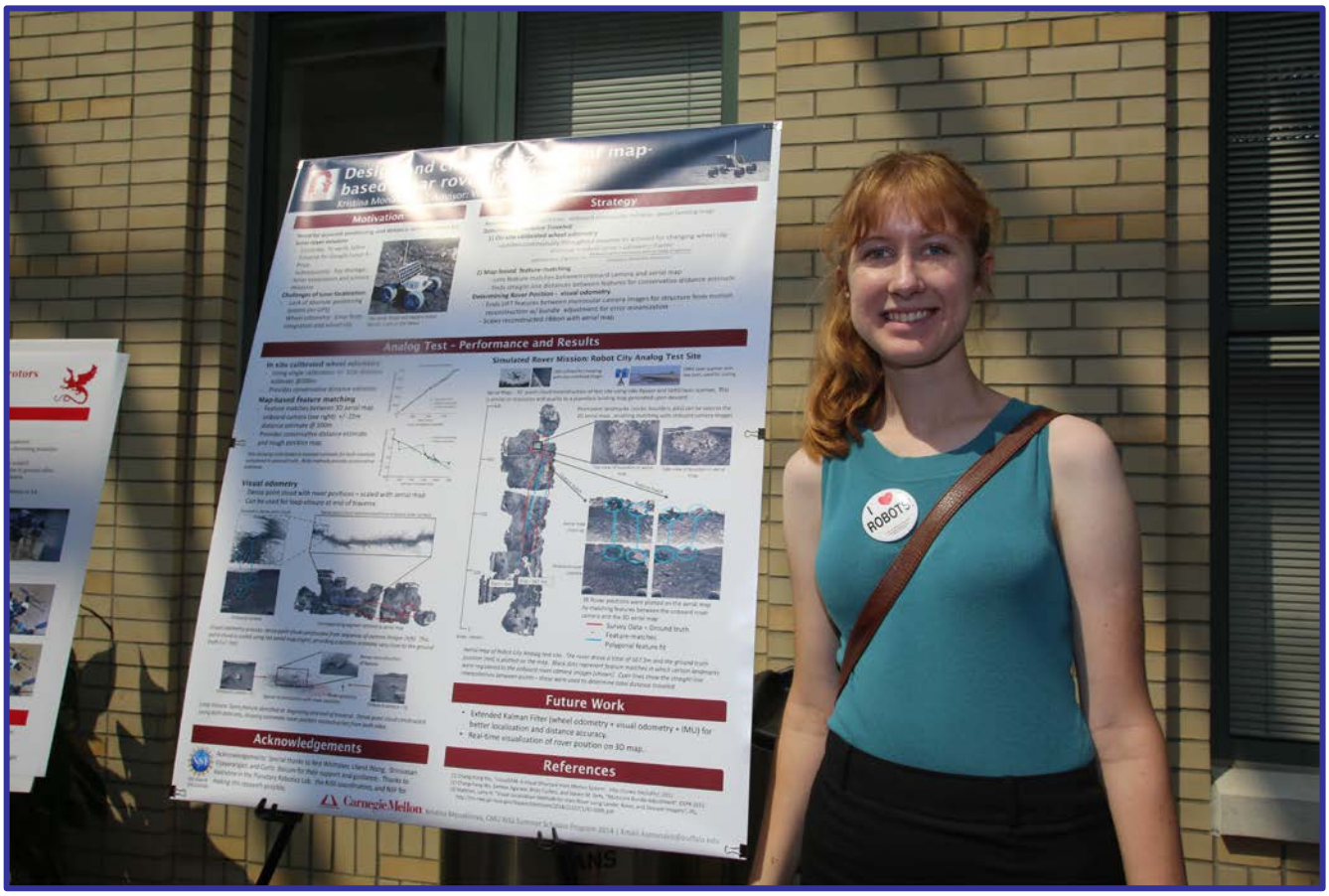
[9] A. Toshev, C. Szegedy, "DeepPose: Human Pose Estimation via Deep Neural Networks," *CVPR*, 2014.

[10] L. Breiman, "Random Forests," Machine Learning, 2001.

[11] T. Sharp, "Implementing Decision Forests on a GPU," *ECCV* 2008.

Kristina Monakhova

RISS 2014



Rover Localization and Distance Verification Using a Planetary Landing Map

Kristina Monakhova
University at Buffalo, the State University of New York
Buffalo, New York 14260
Email: ksmonakh@buffalo.edu

Mentor: William "Red" Whittaker
Carnegie Mellon University
Pittsburgh, PA 15213
Email: red@cmu.edu

***Abstract*—Accurate rover localization and distance determination is necessary for purposeful planetary exploration and scientific missions. In this paper, the use of a planetary landing map is examined for the purpose of lunar localization and total distance traveled measurement in the context of the Google Lunar X-Prize. A methodology for lunar localization using monocular visual odometry scaled with wheel odometry is presented. In this methodology, feature matches between the aerial map and the on-board rover camera are utilized for absolute localization. This methodology was tested in a 500m analog field experiment, during which the rover drove over 500m in lunar-like terrain. This experiment was analogous to the 500m lunar treks that Google Lunar X-Prize teams will be undertaking. Next, the methodology was tested and analyzed in a shorter 100m analog experiment with a more complete data set in Pittsburgh, PA. The results of the test are presented, showing distance determination to within 5 meters and overall localization to within 10 meters.**

***Keywords*—Rover localization, Monocular Visual Odometry, Google Lunar X-Prize.**

I. Introduction

There is a need for simple and robust localization on the lunar surface for purposeful exploration and scientific missions. Recently, the Google Lunar X-Prize has sparked a number of novel lunar rover missions. In addition to requiring localization on the lunar surface, these missions will require precise distance determination to prove with high confidence that they have traveled at least 500m to win the prize. Unlike on Earth, satellite-based GPS localization is not available on the Moon, making absolute positioning challenging.

The Mars Exploration Rovers utilized stereo visual odometry paired with wheel odometry and an IMU (inertial measurement unit) to achieve high accuracy localization [1]. By itself, wheel odometry is cheap and simple to implement; however, it is prone to accumulating error caused by wheel slip. This causes a position estimate with unbounded error over time. Similarly, IMUs are susceptible to sensor noise and drift, causing errors in the position estimate. By using stereo visual odometry, the Mars Exploration Rovers were able to correct for wheel odometry errors caused by wheel slip and generate position estimates to within 1% of the actual rover position [1].

Stereo-visual odometry has proven to be successful for localization; however requires a stereo vision system that is mechanically complex and may be impractical for smaller rover missions. Other techniques employ additional, often expensive, sensors for localization. A popular technique includes utilizing LIDAR-based horizon matching for long range positioning with a several kilometer range [2]. Another technique includes using an upwards facing star tracker for celestial navigation [3]. These strategies have been shown to produce rover localization accurate to within tens of meters; however they require additional hardware, adding system complexity and cost.

Fig. 1. Andy 2 rover exploring a simulated lunar pit in the Robot City test site. The Andy rover mission aims to explore the Lacus Mortis pits on the Moon as part of the Google Lunar X-Prize.

With recent advances in monocular visual odometry and the development of freeware, such as Visual Structure From Motion (VisualSFM), it is possible to generate dense and comprehensive 3D point clouds from monocular camera images [4]. The generated 3D point clouds provide information about the shape of the camera's motion throughout a traverse; however, unlike stereo visual odometry, monocular visual odometry does not provide any absolute scaling or distance information. Recently, there has been research in utilizing inertial measurement units to infer scaling for ground based robots and UAVs [5]. In addition, crude scale can sometimes be inferred from landmarks and known features within the surroundings [5]. Wheel odometry can also be utilized to infer the scale of the 3D point clouds generated through monocular visual odometry.

Recently, there has been active research in robot localization using Google Street View and aerial imagery from Google Maps [6]. Analogously, rover position can be determined from an aerial map generated during landing. Most contemporary planetary landers are equipped with down-facing cameras and LIDAR for autonomous landing operations. Images taken from

a planetary lander during descent could be merged to generate a 3D landing map using structure from motion techniques. The coverage and detail of this map depend on the field of view of on-board sensors and the landing path, however several kilometers of coverage could be expected. This 3D landing map could be utilized for absolute rover positioning on the lunar surface for short-distance lunar missions.

In this paper, the use of a planetary landing map in conjunction with a monocular camera and wheel odometry is examined for the purpose of lunar distance verification and localization. VisualSFM, a freeware software, is utilized for generating 3D point clouds from the on-board rover camera images (rover ribbons). This rover ribbon is scaled with known time-stamped distances from wheel odometry distance estimates. The rover ribbon is then matched to a 3D aerial map that is generated during rover landing. Wheel odometry is utilized for distance verification and is cross-checked to the aerial map for higher statistical confidence of total distance traveled. This system uses basic sensors that are found on most robots (camera, wheel encoders) paired with a detailed landing map that is expected in planetary missions to achieve high performing localization.

The aforementioned strategy was tested in two field experiments during which the rover navigated through analog lunar terrain (Figure 1). Both experiments took place in the Robot City Site in Pittsburgh, PA. During the first experiment, the rover drove over 500m in lunar-like conditions, mimicking the operational conditions of an actual lunar mission. For the second experiment, the rover drove a closed-loop of over 100m in lunar-like terrain, exploring several lunar-like features and returning to the simulated landing site. Results from these experiments demonstrate high confidence in total distance traveled using an in-situ wheel-odometry calibration factor. In addition, a rover localization utilizing visual odometry with matching to the aerial landing map was demonstrated for the purpose of absolute positioning on the moon.

The aim of this research is to provide a framework for a simple, yet accurate lunar localization methodology to enable small-scale lunar rover missions, particularly for Google Lunar X-Prize teams. A distance determination and localization strategy using only an on-board monocular camera, wheel odometry, and an aerial landing map is presented.

II. Field Data

Data was collected in the Robot City Lunar analog test site in Pittsburgh, PA. This site is an excavated construction site with loose dirt, rocks, boulders, and steep pit-like excavation areas. The site's prominent positive (rocks) and negative (pits) features make it an ideal lunar-analog site, since these are the sorts of features that are prevalent on the lunar surface.

Before each of the field tests, an aerial landing site was generated using a multi-rotor UAV (unmanned aerial vehicle). The UAV collected several thousand fly-over images from above the test-site. In addition, a FARO Focus 360 laser scanner was utilized to obtain several high-precision laser scans of the entire test site. The FARO laser scanner was placed at several different locations to capture the entirety of the site. The positions of the FARO scanner were recorded using survey equipment accurate to 5mm in order to reconstruct the area. The UAV images were fusing using VisualSFM to generate a dense 3D point cloud of the site. This point cloud was scaled and aligned using the laser scans. The 3D aerial map was degraded to resemble the quality of imagery expected from the Astrobotic lander (Figure 2).

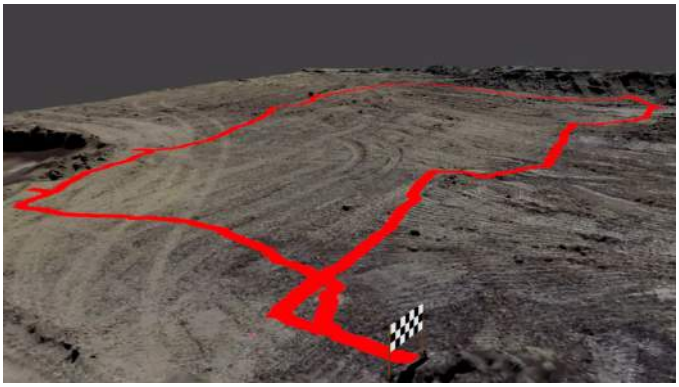


Fig. 2. Aerial Map of analog test site. Ground truth rover position is overlaid in red. The positive and negative features present in the test site are characteristic to the lunar terrain.

Survey equipment was utilized to monitor the ground truth rover position to within +/-5mm by using a tracking prism mounted on top of the rover. This recorded the time-stamped, ground-truth position of the rover for the duration of each experiment. All further analysis of rover position and distance traveled utilizes this survey data for comparison.

During the first test, the rover drove over 500m. Drivers had no prior knowledge of the test site and navigated the rover with an imposed 5 second delay (Figure 8). During the second test, the rover drove a closed-loop of over 100m on the lunar-analog terrain. Rover commands were sent from a mobile ground station with a simulated control delay of 5 seconds. Images were streamed from the rover to the ground station for processing at a rate of 1Hz. These images were then rectified to account for distortions in the camera lens using pre-determined camera parameters. All data was time-stamped for correlation between different data sets.

III. Methodology

A. Wheel odometry

Wheel encoder data is used to monitor the total distance traveled and estimate rover pose. The rover distance and position estimate is susceptible to error caused by wheel slippage. This is mitigated by utilizing an odometry factor calibrated for the lunar terrain.

The odometry factor (distance/turn) is calibrated on site by driving the rover between two features of a known distance apart and recording the number of wheel turns between the features. In the absence of distance information between features, the odometry factor calculated during rover testing can be utilized in place of a lunar calibrated odometry factor. The distance the rover travels is given by:

$$d = t_{\text{wheel}} \cdot s \tag{1}$$

Where t_{wheel} is the number of wheel turns and s is the odometry factor, a factor predetermined in testing that is dependent on the rover wheel diameter and terrain (Figure 3).

For the first test, the wheel odometry factor was calibrated on-site by driving between two rocks visible in the aerial map and measuring the number of turns between those rocks using wheel encoders. The rover path was greater than the straight-line path between the rocks due to object avoidance, resulting in an underestimated wheel odometry factor, which provided underestimates of total distance traveled.

For the second test, the wheel odometry factor was calibrated using the test data from the first test, resulting in wheel distance estimates close to the ground truth.

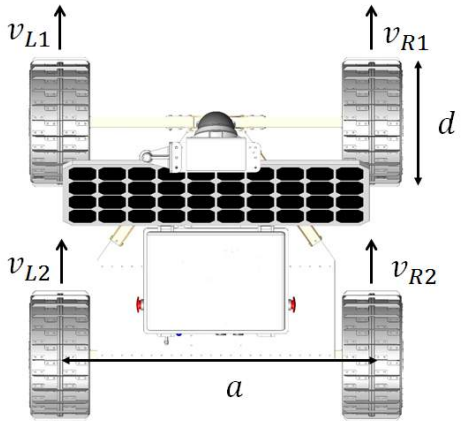


Fig. 3. Andy 2 rover with associated dimensions: d - rover wheel diameter, a - track length, v_{L} - average velocity of left wheels, v_{R} - average velocity of right wheels. Wheel turns were recorded for each wheel during testing.

B. Visual odometry

Visual odometry utilizes feature matches between corresponding image sequences to reconstruct the rover path. Images from the rover's onboard camera during the trek were stitched together to create dense point clouds of the terrain that the rover drove across. Structure from motion (SFM) imaging techniques were used to generate the three-dimensional point clouds and to extract the rover camera positions at each location. Off the shelf software called VisualSFM was utilized for SFM. VisualSFM first employs Scale-invariant feature transform (SIFT) algorithm to detect correspondences between features. These features are tracked between frames and used to recover structure. The camera parameters (focal length, image center, and radial distortion) were utilized in the generation of the camera positions. In order to limit computational time and minimize false matches when searching for correspondences, each image was compared only to its adjacent two neighbors rather than the whole image data. A bundle adjustment procedure was used to distribute errors along the path. This technique generated a sparse point cloud of the rover path with corresponding camera locations. Finally, CMVS/PMVS was utilized to generate dense point clouds of the rover traverse [7] (Figure 4, 9).

For the second test, a framework for gathering transmitted images and running the images through VisualSFM in real-time was developed in Python, however was met with limited success.

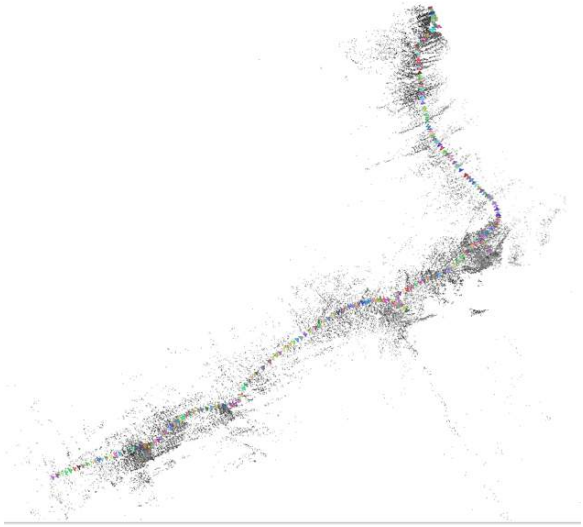


Fig. 4. Sparse rover ribbon with camera positions plotted generated from VisualSFM

C. Scaling

The monocular visual odometry provides a geometry for the rover traverse, but is incapable of determining speed. Wheel odometry distances were utilized to find a scale for the visual odometry point cloud based on corresponding time stamps between the data (Figure 9).

The rover ribbons diverged and generated new models with non-uniform scaling in areas without prominent features. These models were matched together based on time and scaled using timestamped wheel odometry distances (Figure 10).

D. Matching

The rover position can be manually or automatically matched to the aerial map. Matches were found between the aerial map and rover ribbon. These matches serve as landmarks to register the rover ribbons to the 3D aerial map.

During the first experiment, the rover was manually matched to the aerial map. Operators can manually match features from the 3D aerial map to features seen from the onboard rover camera to determine the rover location. 38 features were selected from different points in the rover traverse. Features were selected solely using the transmitted rover images and aerial map - the operator never saw the test site, just like during lunar operations. Since the aerial map is accurate to within 0.1m, absolute rover positions can be determined at each feature to reconstruct a map of the rover traverse (Figure 8).

The point clouds from the rover ribbon and the aerial map were manually aligned in MeshLab though feature matching. This process could be automated using iterative closest point (ICP) algorithms.

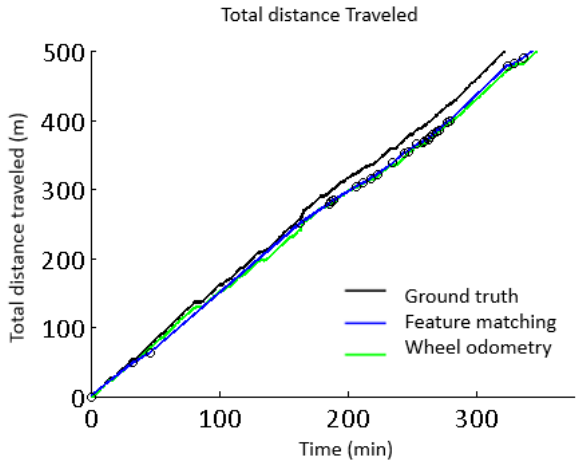


Fig. 5. Total distance traveled estimate for both wheel odometry and manual feature matching during the 500m test.

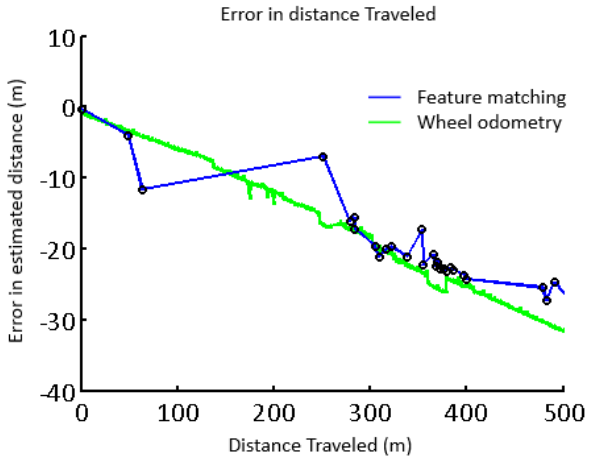


Fig. 6. Error in the distance traveled estimate over the total distance traveled for both wheel odometry and manual feature matching during the 500m test.

IV. Results

A. Test 1 - 500m trek

The distance verification analysis utilized 573.95 meters of rover traverse. From this, the wheel odometry estimated that the rover had traveled a total of 537.1 meters and the feature recognition estimated a total of 539.5 meters. From wheel odometry alone, the standard deviation for the estimate of total distance traveled is 10.3 meters (Figure 5) (Figure 6).

Using feature recognition from the aerial map, the position estimate at each feature has a maximum error less than 8.1 meters. The error in the feature position increases for segments without any prominent features and decreases for segments where there are copious features and it is easy to make visual matches. The estimate of distance traveled using feature matches is based on the linear interpolation between features, assuming the minimum straight line distance between points. Due to this, the distance estimate is always less than the total distance traveled within the bounds of the aerial-map error, since the actual rover path is not typically a straight line. Once the total distance traveled from feature matching equals 500m, there is a high level of confidence that the rover has traversed at least that distance (Figure 8).

B. Test 2 - 100m trek

The wheel odometry was able to provide distance estimates to within 5 meters of the true distance throughout the length of the traverse (Figure 7). This distance estimate was utilized to scale the visual odometry ribbon. The position estimate was able to determine the rover absolute position to within 10 meters (Figure 10).

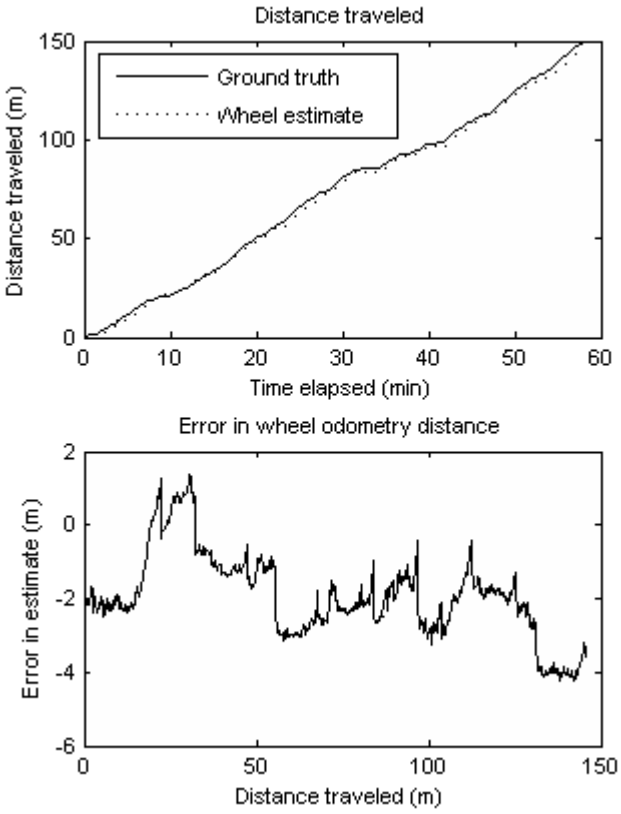


Fig. 7. Error in wheel odometry over a 100m rover trek.

V. Conclusion

During the first field experiment, distance verification was accomplished using wheel odometry and manual feature-matching to an aerial map. These two systems provide redundancy for distance verification. Rover position was estimated using feature matched and straight-line interpolations between matches.

During the second field test, the total driven distance was determined to within 5m, and the rover position was calculated to within 10m. This was accomplished using visual odometry from a monocular camera scaled with wheel odometry. A complete scaled rover ribbon was constructed for the duration of the test, however was not matched to the aerial map due to time limitations.

Throughout this paper, a simple rover localization system

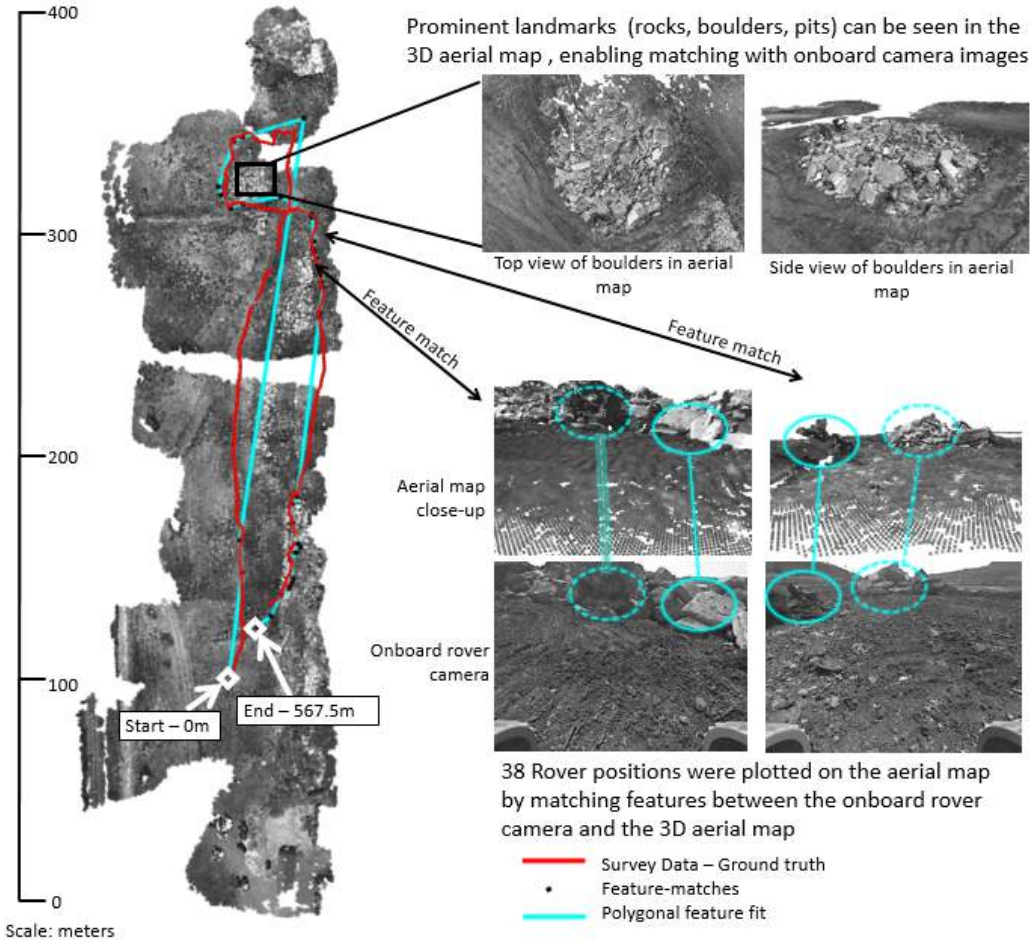


Fig. 8. Aerial map of Robot City analog test site. The rover drove a total of 567.5m and the ground truth position (red) is plotted on the map. Black dots represent feature matches in which certain landmarks were registered to the onboard rover camera images (shown). Cyan lines show the straight-line interpolations between points these were used to determine total distance traveled.

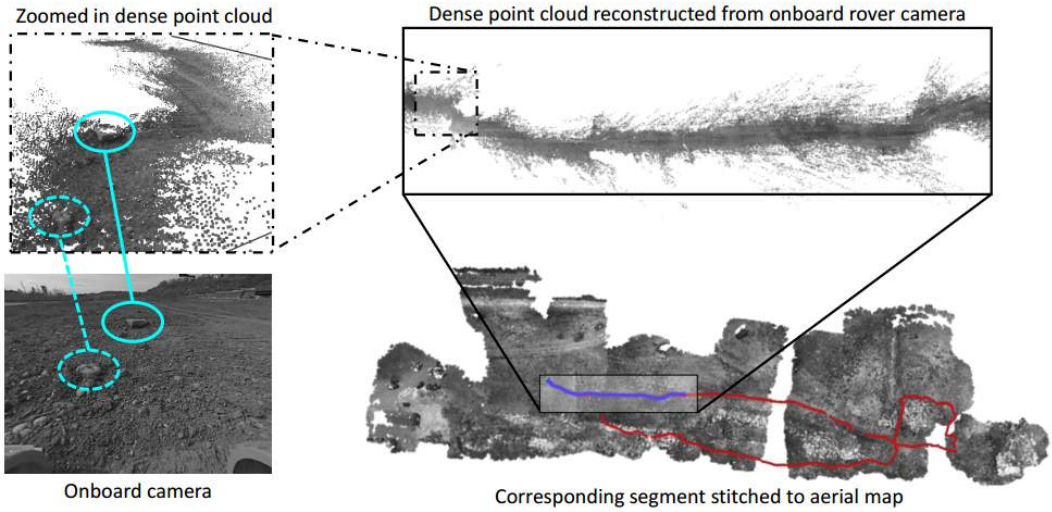


Fig. 9. Visual odometry process: dense point cloud constructed from sequence of camera images (left). This point cloud is scaled using the aerial map (right), providing a position estimate very close to the ground truth (+/- 5m).

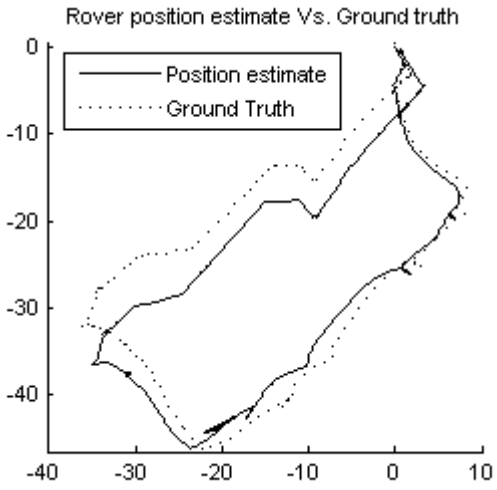


Fig. 10. Rover position estimate Vs. Ground truth for 100m trek. The position estimate was generated by fusing the camera positions from VisualSFM and scaling the rover ribbon using wheel odometry.

without the need for additional hardware other than a camera and wheel encoders was presented.

VI. Future Work

Next, the localization methodology presented in this paper needs to be automated to enable real time rover localization. In addition, a Kalman filter should be implemented to provide a statistically optimal position estimate. Custom feature detection algorithms specially tailored to the lunar surface need to be developed to enable better feature matches between image frames. This will enable to the generation of a more accurate rover ribbon for localization. A methodology for automatically matching the rover ribbon to the aerial map needs to be investigated. Automatic matching will enable full GPS-like absolute positioning on the lunar surface.

This paper aims to serve as groundwork for a lunar localization for the Astrobotic Google Lunar X-Prize mission. Further work is necessary to implement this methodology on the ground station for real-time distance and position monitoring.

Acknowledgment

I would like to sincerely thank Red Whittaker, Uland Wong, Srinivasan Vijayarangan, Curtis Boirum, and the rest of the Planetary Robotics Laboratory at Carnegie Mellon for their support and assistance. In addition, I would like to thank Rachel Burcin and the RISS program for making this research possible.

References

[1] Y. Cheng, M. Maimone, and L. Matthies, Visual Odometry on the Mars Exploration Rovers, 2005.

[2] P. J. F. Carle and T. D. Barfoot, Global Rover Localization by Matching Lidar and Orbital 3D Maps, pp. 881886, 2010.

[3] D. A. Sigel and D. Wettergreen, Star Tracker Celestial Localization System for a Lunar Rover, pp. 28512856, 2007.

[4] C. Wu, Towards Linear-Time Incremental Structure from Motion, 2013 Int. Conf. 3D Vis., pp. 127134, Jun. 2013.

[5] J. Hu, C. Tseng, M. Chen, and K. Sun, IMU-Assisted Monocular Visual Odometry Including the Human Walking Model for Wearable Applications, pp. 28942899, 2013.

[6] S. Muramatsu, T. Tomizawa, H. Matsuda, S Kudoh, T. Suehiro, "Mobile robot localization technique using Web-based aerial photos," Mechatronics and Automation (ICMA), 2012 International Conference on , vol., no., pp.1886,1891, 5-8 Aug. 2012 doi: 10.1109/ICMA.2012.6285109

[7] C. Wu. Visualsfm: A visual structure from motion system. 2011.

[8] Y. Gao, C. Spiteri, M.-T. Pham, and S. Al-Milli, A survey on recent object detection techniques useful for monocular vision-based planetary terrain classification, Rob. Auton. Syst., vol. 62, no. 2, pp. 151167, Feb. 2014.

Cristina M. Morales Mojica

RISS 2014

Visual Programmer Converter for Untethered Running of the Hummingbird Duo

Cristina M. Morales Mojica, *University of Puerto Rico at Bayamón, Puerto Rico,* and
Dr. Illah R. Nourbakhsh, *Carnegie Mellon University, Pittsburgh, Pennsylvania*

***Abstract*— The Hummingbird is an electronic kit designed for educational purposes. It has been a tool for teachers to integrate robotics into their classrooms and to encourage students to pursue STEAM fields. While the Hummingbird has had positive impact on robotics education, the device is limited by requiring a wired connection to a computer so that it is able to run the application. In response to this limitation, the new Hummingbird Duo has the ability to run untethered by changing the controller to Arduino Mode. However, to use this feature some programming knowledge is needed, and our major targeted users (students and teachers) often have limited programming experience. We propose a converter that will simplify the process of coding. We use the CREATE Lab Visual Programmer's output to generate an equivalent program for the Arduino. The converter can be used as a standalone application and will also be integrated in the Visual Programmer. This will remove the need for the user of writing the Arduino code, and can also be used as a tool for learning programming.**

***Index Terms*— educational robotics, creativity, STEAM, technology fluency, converter, Arduino, controller, embedded systems.**

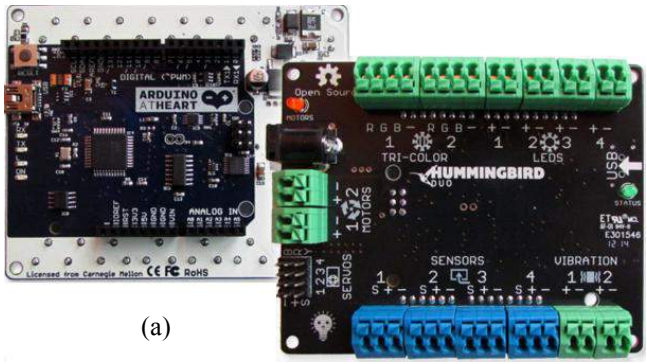
I. Introduction

The Hummingbird is an electronic kit designed for educational purposes, and has been a tool for teachers to integrate robotics in their classrooms. It also encourages students to pursue fields in programming, engineering and robotics [1][2]. Although this device has made an impact in ordinary classrooms by inspiring kids to explore the world of robotics, the device has to be tethered at all times when executing a program. This means a computer with the required program is necessary to use this kit. This is a problem for schools that have a limited amount of computers.

This educational kit contains a controller (see Figure 1) that is easy to use and program. Users can manipulate LEDs, motors, servos and sensors by programming the device using the CREATE Lab Visual Programmer. The Visual Programmer was developed by the Community Robotics, Education and Technology Empowerment Lab (CREATE Lab) at Carnegie Mellon University and was designed to help the user to program the Hummingbird controller [4]. As other visual programming tools, this allows the user to commit logical errors, but is not possible for the user to have syntax error.

The CREATE Lab Visual Programmer's main components are expressions and sequences. Similarly to a video which consist of a combination of images; sequences are a combination of expressions. Expressions can be defined as the basic unit of the Visual Programmer. Every other component available will be a set of expressions. Creating an expression is designing a basic action or a state that the Hummingbird can take such as turning on an LED, moving motors at a specific speed or moving a servo to a certain angle. It can also have more than one output. For example, one state can be to turn the LED on and at the same time move a motor (Figure 1 is an example of an expression with multiple outputs). Sequences are more complicated structures, in which the user can add a list of expressions, conditional loops, counters, and other sequences. Conditional loops are decision making components that can be set to repeat a set of elements and counters are structures that repeat a set of elements a number of times defined by the user. All of these components will be discussed in more detail in a further section.

With the new Hummingbird Duo, applications can be run in the device without a connection to the computer. This new controller has an Arduino Leonardo microcontroller integrated into the board, which allows the user to write the program in the Visual Programmer, as in previous versions, or write the program in Arduino [3]. When the user writes a program for the Arduino, also known as a *sketch*, the code can be uploaded to the controller and the program may run untethered. Using this method, the controller changes from Hummingbird Mode to Arduino Mode. This eliminates the need of the computer when running the program. However, the Hummingbird Kit has mainly been used by users that have no previous background in programming, and learning how to code can be a long and frustrating process. The lack of knowledge in the subject could be a barrier for using this new feature. This is why we have created a converter that will facilitate the process for the user by taking the program developed with the CREATE Lab Visual Programmer and creating an equivalent in Arduino code.



(a) (b)

Figure 1. New Hummingbird Duo. (a) Back of the controller, where the integrated Arduino can be seen. (b) Front of the controller.

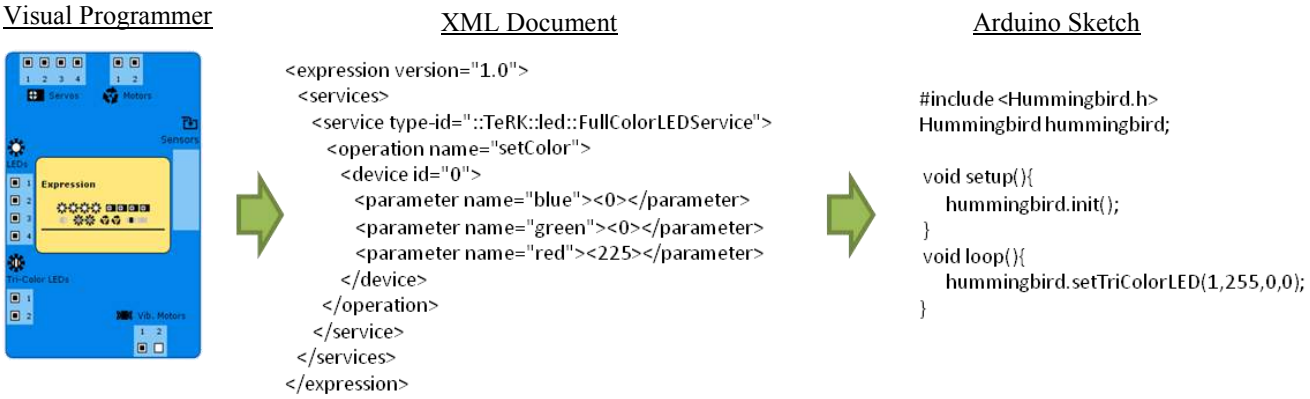


Figure 2. Process of conversion. The user creates the application using Visual Programmer, and this program is saved as an XML file. The converter uses the XML file to search for necessary information and translate it to an equivalent program for the Arduino Sketch.

Even though our major target are users with no previous experience in programming, the converter will be beneficial to both newcomers and people already knowledgeable about the subject. Beginners will have the option to use the Hummingbird Duo untethered without the need of knowing how to program, but also it will give the user an opportunity to examine the code provided, learn from it, and make an easier transition from novice to intermediate. Users with experience in programming can use this tool to speed up the process of creating the code, giving them a good base from where they can build their application and create more complicated programs.

II. Development of the Converter

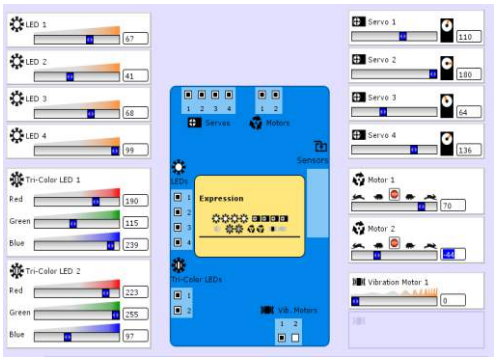
When using the CREATE Lab Visual Programmer, the user develops a program, a *sequence*, that is saved as an XML document. The converter, developed using Java, uses an XPATH library to search through the XML document for necessary information for the final Arduino sketch (see Figure 2). The information retrieved from the XML document is translated to equivalent statements of Arduino code that use the Hummingbird Library. The equivalent statements are then saved as an Arduino sketch. The user can upload the final output to the Hummingbird Duo, and run the application directly off of the microcontroller.

The Visual Programmer provides the user with four different components: Expressions, Conditional Loops, Counters, and Sequences. The following section will present in detail how we developed the translation from each Visual Programmer component to its equivalent statement for the Arduino code.

A. *Expressions*

With expressions, the user has the option to convert an expression by itself (as shown in Figure 3) or as a component in a sequence. When expressions are part of a sequence, each expression is written in the Arduino sketch as a method. These methods, the converted expressions, will be located below the *loop* method (equivalent of a main method). Inside the *loop* method, the converter will insert the method's call that invokes the corresponding expression statements.

Through the conversion of a sequence, every time a new expression appears, the converter translates it to the equivalent code, saves this code in a list, and writes the method's call. On the other hand, when an expression is repeated in a sequence, the converter writes in the Arduino sketch the calling of the corresponding method without converting it again and continues to the next element. Each expression will be displayed as a method to avoid repetition and for better understanding of the program's flow. It is also a way to teach the concepts of methods to newcomers to programming. Finally, if an expression was deleted, the user has the option of continuing with the conversion without the missing file, or interrupting the conversion.



(a)

```
hummingbird.setLED(2,105);
hummingbird.setLED(1,171);
hummingbird.setLED(4,252);
hummingbird.setLED(3,173);
hummingbird.setServo(1,134);
hummingbird.setServo(3,78);
hummingbird.setServo(4,165);
hummingbird.setServo(2,219);
hummingbird.setVibration(1,0);
hummingbird.setTriColorLED(2,223,255,97);
hummingbird.setTriColorLED(1,190,115,239);
hummingbird.setMotor(2,-112);
hummingbird.setMotor(1,179);
```

(b)

Figure 3. The expression component. (a) The view for designing an expression in Visual Programmer. (b) The equivalent statements of the expression (a) for the Arduino.

B. *Conditional Loops*

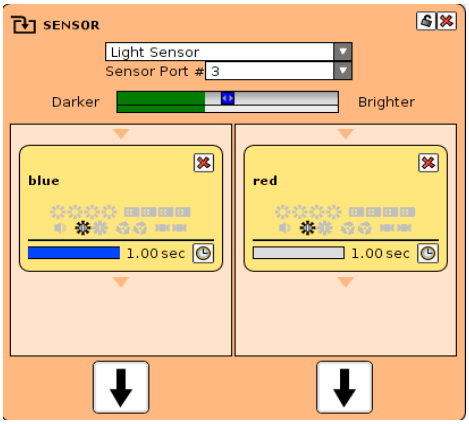


Figure 4. Conditional loops are decision making components that can be set to repeat the elements inside the square by changing the arrows below. The conditional loop can generate four different possibilities that translate to if-else statement or while statement.

Conditional loops are decision making structures associated with a specific sensor. The conditional acts on a threshold value specified by the user and the information received by a sensor. This element is treated as an *if - else* statement or a *while* statement, and for better understanding, in this paper we will label the left square as the *if* and the right square as the *else* (see Figure 4). In both of the squares the user can add expressions and sequences. Also, the set of elements inside of the squares may be repeated by changing the arrows below them.

1) *Regular **if-else** statement*

This option will generate a regular *if - else* statement where the instructions or set of instructions will execute at most once and move on to the next element. (see Figure 5a)

2) ***if** with repetition and **else** without repetition*

In order to have the corresponding behavior, it is needed to change the structure from an *if - else* statement into a *while* statement followed by a set of statements. When the user selects the option of repeating the *if* square, it means that it will continue doing the set of elements in its body until the condition is no longer satisfied. In the Arduino Sketch, the *if* square will be written as a *while* containing the corresponding statements. Outside the *while,* there will be the *else* square statements. When both blocks are executed it will continue to the next element. (see Figure 5b)

3) ***if** without repetition and **else** with repetition*

When the *else* square is set for repetition, and the *if* is not; the "less than" sign needs to be changed to "greater than" sign. Also the order of the statements that invoke the methods from the squares need to be changed. The *else* square statement will be written as a *while*, and when the condition is no longer satisfied, the *if* statements will execute. (see Figure 5c)

4) ***if** and **else** with repetition*

This option will generate an infinite loop, which could be what the user intended, or it could be a logical error. Both the converter and the Visual Programmer allow this situation. It will be translated to an infinite *while* loop that has an *if - else* statement inside. (see Figure 5d)

(a) Regular *if-else* statement

```
if (hummingbird.readSensorValue(3) < 511) {
   blue();
   delay(1000);
}else{
   red();
   delay(1000);
}
```

(c) *if* without repetition and *else* with repetition

```
while (hummingbird.readSensorValue(3)>511) {
   red();
   delay(1000);
}
   blue();
   delay(1000);
```

(b) *if* with repetition and *else* without repetition

```
while(hummingbird.readSensorValue(3)<511) {
   blue();
   delay(1000);
}
   red();
   delay(1000);
```

(d) *if* and *else* with repetition

```
while(true){
   if(hummingbird.readSensorValue(3)<511) {
      blue();
       delay(1000);
    }else {
       red();
      delay(1000);
   }
}
```

Figure 5. The four different possibilities when using the conditional loops

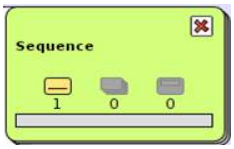
C. *Counter*


```
for(int counter = 0; counter <5; counter++){
    LEDs();
    delay(500);
    Servoat27();
    delay(1000);
}
```

Figure 6. Counters allow the user to repeat a set of components a specified number of times, as in a *for* loop.

In Visual Programmer, the counter is an element that allows the user to repeat a set of expressions and sequences a specified number of times. This is transformed to a *for* statement with the specified number of iterations.

D. *Sequence*




```
//Start Seq: Sequence.xml:
    Expression();
    delay(1000);
        . . .
//End Seq: Sequence.xml
```

Figure 7. Sequences are the combination of all the components, including other sequences.

Sequences are containers of all the previously discussed elements, including other sequences. Because there is no maximum depth of sequences within sequences, we approach it by implementing tail recursion. (see Figure 7)

III. SENSOR VALUES CONVERSION

In this section we will discuss problems that we encountered while testing the sensors. Each sensor had a particular issue which needed a specific approach, and these will be discussed in more detail further in this section. We want to stand out that sensors are only used with the conditional loop component. Also, the flow of program will depend on the threshold value set by the user at the percent bar and the information received from the sensor (see Figure 4). We want to be clear that conditional loops are conditional statements when are translated to the Arduino sketch, and this conditional statement (*if-else* or *while*) will use this threshold to evaluate if the condition is satisfied, and with this decides the flow of the program. This is why it is important to have an accurate and representative value of that threshold from the Visual Programmer. A common problem between sensors was that the threshold value saved in the XML document was the number of the percent. For example, when the user set the threshold at the middle of the percent bar the value saved at the XML document was 50 as in fifty percent, and not the real value. Usually sensors have a range from 0 to 1023, the real value when the threshold is set at 50% should be 511. To be able to have that real value, a percent conversion was applied. After this, we notice specific problems with the potentiometer sensor (rotary knob), distance sensor and temperature sensor. These other situations happened because not all of the sensors have the same range (0 to 1023) and not all have the values in ascending order.

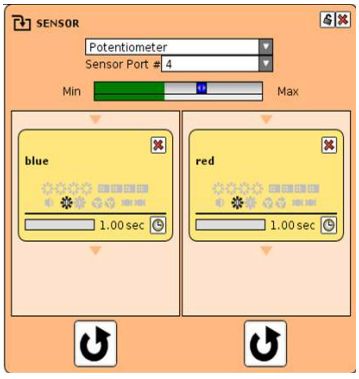


Figure 8. Conditional Loop with threshold in 65%

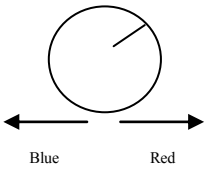


Figure 9. Representation of the potentiometer while running the program in Visual Programmer

```
while(true){

if(hummingbird.readSensorValue(4)>358) {
            blue();
            delay(1000);

}
 else {
            red();
            delay(1000);

            }
```

Figure 10. Correct code that behaves like the original program

A. *Potentiometer (Rotary Knob)*

Figure 11. Hummingbird's Kit Potentiometer.

When running the application with Visual Programmer and with the threshold set at 65% as shown in Figure 8, the potentiometer was leaning towards the right (see Figure 9). With the percent conversion, we expected to have the same behavior when running the program with the converted file, but the mark was leaning towards the left. This is because the potentiometer range is not in ascending order (1023 to 0), the numbers from the left are greater than the numbers from the right. Meanwhile, most of the sensors are from 0 to 1023. To address this situation, an equivalent number conversion was applied. This caused the program to have the correct values, but it did the opposite function. This was solved by changing the "greater than" sign to a "less than" sign (see Figure 10). The distance sensor had the same behavior, and the same approach as the potentiometer was applied, with the difference that the range from the distance sensor is smaller, and its maximum value is 600, not 1023.

B. *Temperature*

Even though the temperature value was not executing backwards, it also needed a conversion. During execution, the value generated with the percent conversion was higher than it needed to be in order to behave as the original program. A special conversion was applied by subtracting 20% from the original value. This made the value more accurate.

IV. Conclusion

After using the converter, the user will have an equivalent program in Arduino code. When the conversion is completed, a window with the new file will appear on the screen for the user to select and use. The converter is available as a standalone application, and is also integrated in the CREATE Lab Visual Programmer. When using the standalone application, the user will search for the XML file that needs to convert. When using the Visual Programmer, the user selects the expression or sequence and presses an export button. Once the user has the Arduino Sketch, the user needs to upload the code to the Hummingbird. When this is saved in the device's memory, the user can unplug the Hummingbird Duo from the computer and run the application.

V. Results

The Hummingbird Duo has not been released and neither has the new version of the CREATE Lab Visual Programmer with the integrated converter. Users with older versions of the Hummingbird will be able to use the new Visual Programmer, but the option of exporting the code is not available for those versions. Since the Hummingbird Duo and the new version of the Visual Programmer have not been released, all testing was done in a controlled environment. It is our next step to do testing in classrooms and with volunteer subjects.

This is a tool for any user that wishes to use the Hummingbird Duo controller in Arduino Mode and does not have the knowledge or the time for developing the code. We wanted to give the users the option and comfort to use this educational kit with no previous programming background. We hope to have eliminated any intimidation or frustration that this lack of knowledge might cause, while encouraging users to search further, and become part of the scientific community.

VI. Acknowledgments

We would like to thank the National Science Foundation (NSF) for supporting this project. We would also like to thank Tom Lauwers of BirdBrain Technologies and the CREATE Lab for their contributions, especially Chris Bartley, Jennifer Cross and Emily Hamner.

VII. References

[1] I. Nourbakhsh, E. Hamner, T. Lauwers, C. F. DiSalvo, and D. Bernstein, "TeRK: A Flexible Tool for Science and Technology Education," *Proceedings of AAAI Spring Symposium on Robots and Robot Venues: Resources for AI Education*, March, 2007.

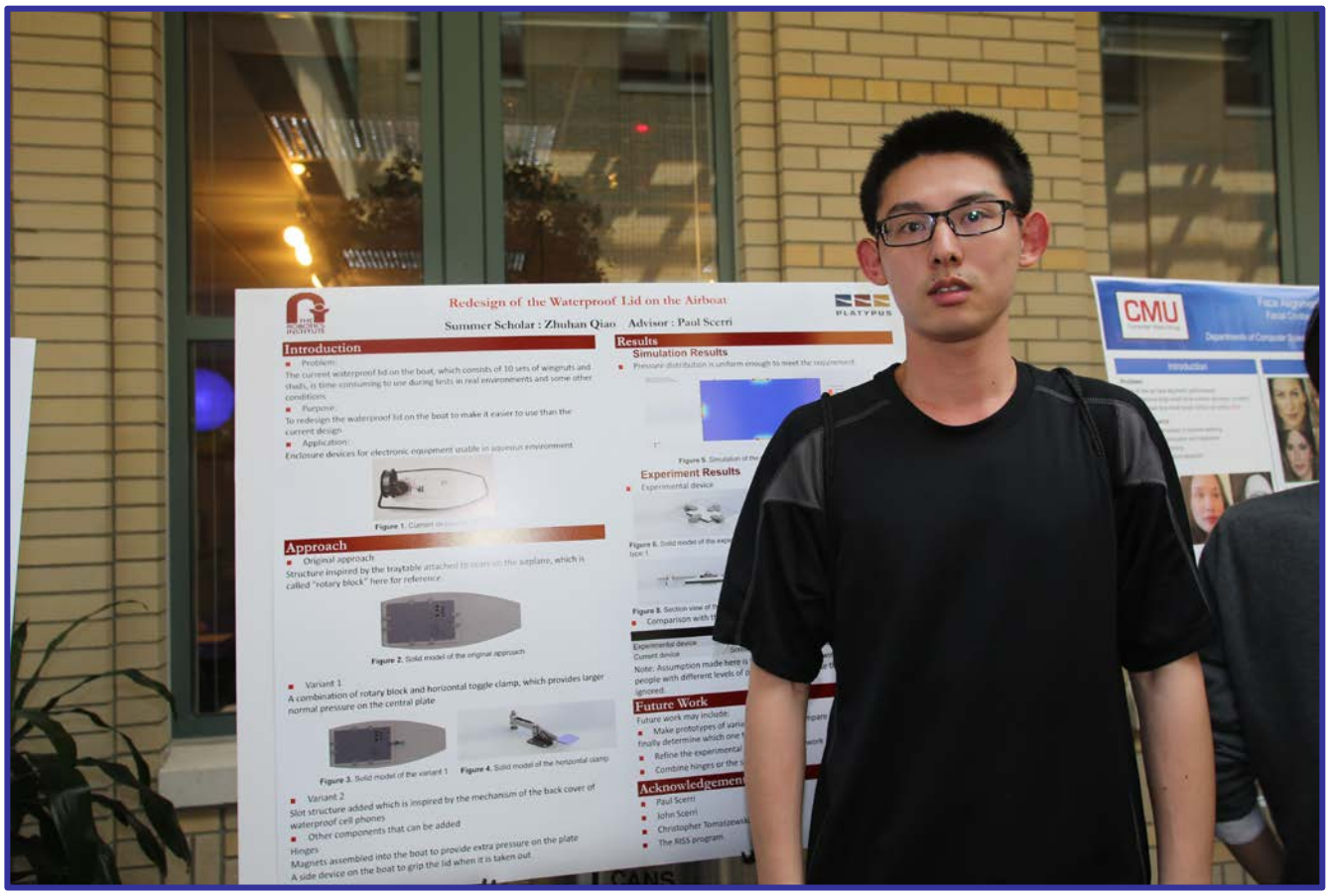
[2] E. Hamner and J. Cross, "Arts & Bots: Techniques for distributing a STEAM robotics program through K-12 classrooms," *Proceedings of the 2013 IEEE Integrated STEM Education Conference (ISEC)* , March, 2013.

[3] T. Lauwers, "Aligning Capabilities of Interactive Educational Tools
to Learner Goals". Ph.D. thesis. Dept. Robotics Inst., Carnegie Mellon Univ., Pittsburgh, P.A, 2010.

[4] J. Cross, C. Bartley, E. Hamner, I. Nourbakhsh, "A visual robot-programming environment for multidisciplinary education," *Robotics and Automation (ICRA), 2013 IEEE International Conference on* , vol., no., pp.445,452, 6-10 May 2013
doi: 10.1109/ICRA.2013.6630613

Zhuhan Qiao

RISS 2014



Redesign of the Waterproof Enclosure on the Lutra Autonomous Boat

Zhuhan Qiao, RI Summer Scholar, Robotic Institute, Carnegie Mellon University

***Abstract*—This paper presents the mechanical redesign of the waterproof enclosure, specifically developed for the new Lutra 1.1 autonomous boat from the company Platypus. The greatest constraint of the current enclosure is its inconvenience to operate and the feature of time-consuming. More concretely, the tedious operation includes screwing and unscrewing 14 wingnuts. Several mechanical simplifications have been made to meet the requirements both for time and ease of operation. In this paper, a detailed mechanical description of the redesigned system is provided, fulfilling the necessary design requirements. The main outcome of this work is a full size experimental device capable of mitigating the current drawbacks, especially decreasing the time consumed to a great extent. The device has been assessed in terms of waterproof tests, operating steps and time consumption, resulting in a rather satisfactory and promising performance.**

***Index Terms*—Lutra, autonomous boat, rotary module, waterproof, mechanical simplification**

[1] Introduction

In recent years, electronic equipment has been used more and more intensively inside autonomous watercraft. As a result, the enclosure device for electronic equipment in aqueous environment has become a key part of the system's mechanical design.

The main function of the enclosure is to secure the water-tightness property. One of the other major benefits is its removability. It allows the operations of opening and closing for setting up and adjusting the internal electronic and other types of devices. The autonomous boat Lutra is shown in Fig.1. The part pointed by an arrow on the top is the lid.

Fig 1. The boat Lutra (source: Platypus Inc.)

Assembled in the boat's hull, the electronics compartment contains an Android smartphone where the main computation is performed, a custom made embedded board that acts as an interface between the phone and other peripherals and finally a LiPo battery that powers the entire system.

Several versions of the enclosure have been made so far by the staff from Platypus. In the initial phase, a bolt-nut mechanism was used along with an electric screwdriver to screw and unscrew all the nuts for each boat. Then, a stud-wingnut mechanism was adopted. And this is also the current method, enabling completely manual operation. Both of the methods are rather time-consuming and inconvenient to operate.

On the basis of the previous approaches, a novel mechanism capable of faster operation has been designed. The new mechanism is inspired from the tray table on the airplane. The rotary block is provided to facilitate the ease of operation.

The paper is organized as follows: Sec.II introduces the design requirements that led to the entire design workflow, Sec.III describes the mechanical design in detail, while Sec.IV presents the development of the experimental devices and reports and analyses of their performances versus the requirements, finally, Sec.V suggests the future work that could be done.

[2] Design Requirements

As mentioned above, three essential requirements are to be met, which are ①the rapidity, ②ease of operation, and ③the waterproofness.

Thus, it should have an easy-to-use design. The dimensions of the mechanism are constrained to the current size of the boat. Rotary blocks are designed to press the lid down to the body of the boat to ensure the waterproofness.

To make the requirements more concrete, they are summarized as the follows:

1. Kinematic: The enclosure can be opened and closed and the lid can be moved vertically. The other parts of the enclosure are fixed relative to the boat body. Rotary blocks can be rotated horizontally in their individual place.

2. Waterproofing: Water cannot enter the compartment when the boat is running.

3. Dimensions: 17 inch long by 14 inch wide is the maximum dimensions of the enclosure according to the current size of the boat. The height of the enclosure needs to be larger than the sum of the thickness of the lid and the gasket.

Zhuhan Qiao is with the Mechanical Engineering Department, Nanjing University of Science and Technology in Jiangsu Province, China. (e-mail: drewhanchiao@gmail.com)

This research was conducted at Carnegie Mellon University under the guidance of Paul Scerri.

[3] MECHANICAL DESIGN

To reduce the complexity of the operation, the previous action of screwing and unscrewing is to be replaced by simply rotating the rotary blocks which are assembled on the protruding edge around the lid.

There are many ways to guarantee waterproofness. In this work, the use of a water-resistant gasket was chosen because of its advantage of not needing complicated manufacturing process compared to other ones.

[4] Main Design Elements

The Base

The base is the skeleton or the platform of the enclosure where all other components are inserted.

The Rotary Block

The rotary block is the major component of the enclosure mechanism that transmits the pressure onto the top of the lid.

Its shape (Fig. 2) is designed to have a round contour along the edge on the lower half plane. This feature increases the smoothness during operation.

Fig. 2 A solid model of the rotary block showing its shape

Sleeve

The sleeve is the component that creates space in vertical direction allowing free rotation for the rotary block. The thickness of the sleeve should be slightly larger than that of the rotary block.

Rotary Module

This module (Fig. 3) mainly consists of a rotary block, a sleeve, a screw, a t-nut and washers. It can be inserted into the system of the waterproof enclosure.

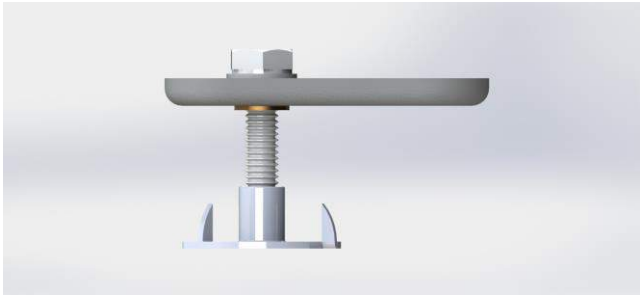


Fig. 3 A solid model of the rotary module

The Lid

The lid is the module that acts as a mechanical interface among the sensors, electronic boards, the smartphone and the battery. The current lid is to be used, which is shown in Fig. 4.

The Gasket

The gasket is the key component that guarantees the waterproofness.

The Handle

The two handles (Fig. 4) which are attached on the top of the lid are used for taking out and putting in the lid.

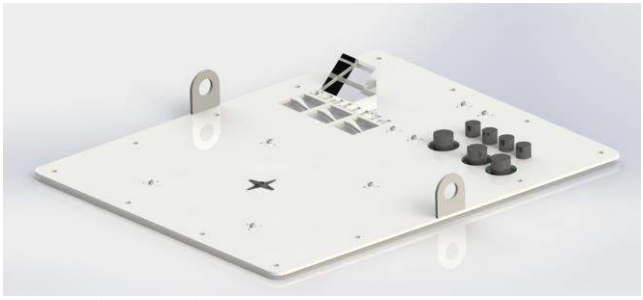


Fig. 4 A solid model of the lid and the handles (source: adapted from the drawings from Platypus Inc.)

[5] Fabrication of the Components

Most of the pieces composing the new waterproof enclosure were fabricated using standard procedures employing one or more of the following manufacturing processes:

[6] Laser cutting
[7] 3D printing
[8] Sawing
[9] Gluing

[10] EXPERIMENT RESULTS AND PERFORMANCE ANALYSIS

The mechanical design presented in Sec.III is the outcome of a conceptual design based on the technical requirements, as reported at the beginning of the paper. Afterwards, experimental devices were built for tests to validate the design. The description of two different sizes of experimental devices are reported first, followed by the demonstration of the experiments and obtained results.

Experiment Setup

In order to reduce cost, a scaled down experimental device (Fig. 5) was made first according to the ratio of 1 to 5, followed by a full size one (Fig. 6). Around the peripheral of the full size device, multiple rotary modules can be inserted to enable testing of different numbers and distributions of them.

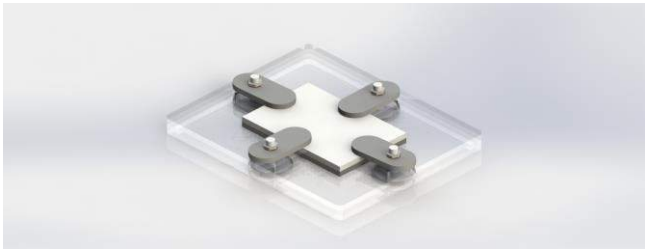


Fig. 5a A solid model of the scaled down experimental device

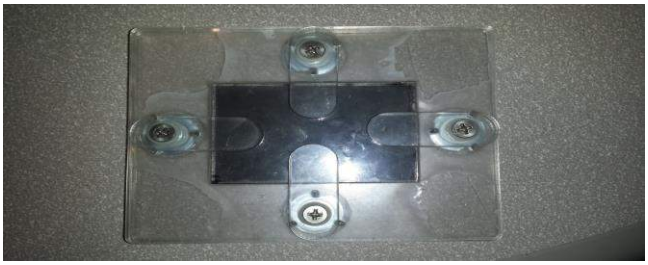


Fig. 5b Actual hardware built of the scaled down experimental device

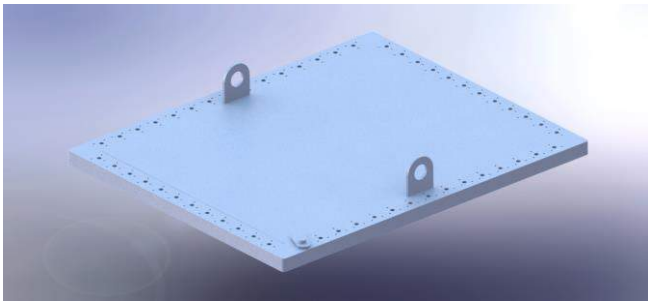


Fig. 6 A solid model of the full size device

The cross-sectional view of the scaled down experimental device is shown for exposing the inner structure (Fig. 7). The specific mechanism in the full size device is analogous to the scaled down one.

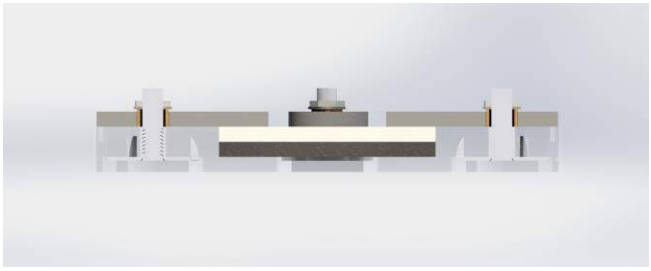


Fig. 7 Cross-sectional view of the scaled down device

For the purpose of assessing the performance of the experimental devices, waterproofing tests were adopted to test the waterproofness. A piece of dry absorbent tissue is placed under the lid to check if the tissue is wet or not after the test. The test consists of 2 steps:

[11] Splash water onto the device to simulate the water environment

[12] Directly current impact on the lid

After this, a comparison of the time consumed between the experimental device and the current enclosure was made.

Experiment Results

The results of the scaled down device showed that it passed the waterproof test. Thus, a full size device was built to be tested next. After several trials of different combinations of rotary modules, the one (Fig. 6) with the following distribution of rotary modules passed the waterproof test.

Fig. 6 Actual full size experimental device that passed the waterproof test

The table below shows the obtained time consumed by different devices. Time is measured by a stopwatch.

	Operations required	Time consumed
Full size device	Turn 12 rotary blocks	around 25 seconds
Current device	Screw 14 wing nuts	around 2 minutes

Table 1. Comparison between the full size and current device

The results showed a around 75% speed improvement.

[13] Future Work

Future work could focus on:

[14] Making modifications to the shape of the edge of the lid and try other washers and gasket to further smooth the operation.

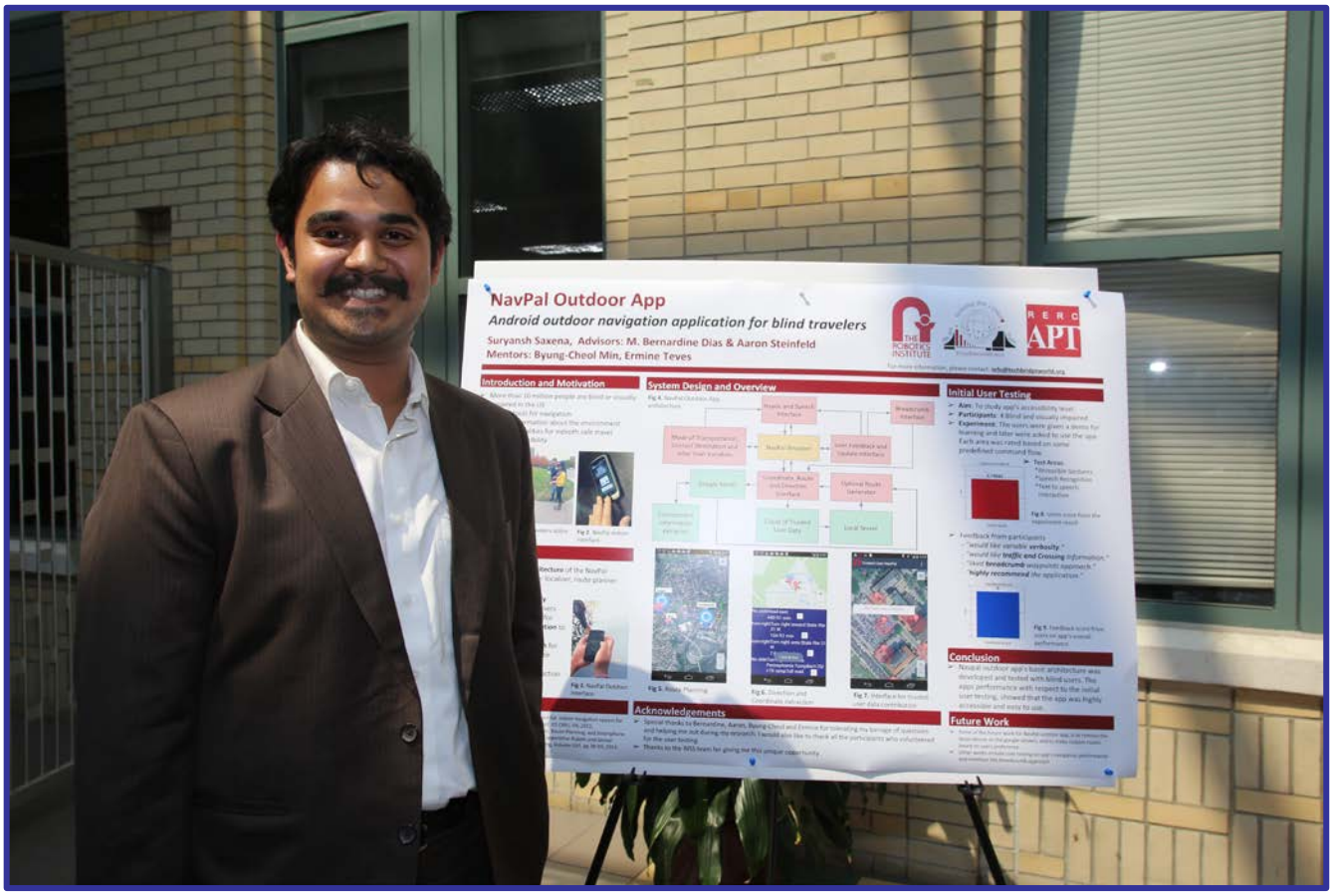
[15] Making slight changes and then putting the full size device onto the boat

Acknowledgments

The author would like to thank Prof. Paul Scerri for his guidance, all the other graduates in the team for upholding the tests and Rachel Burcin and other coordinators for organizing the RISS program throughout the process.

Suryansh Saxena

RISS 2014



System and Architecture Design of NavPal Outdoor Navigation Aid for Blind and Visually Impaired Users

Suryansh Saxena*, Byung-Cheol Min†, M.Bernardine Dias† and Aaron Steinfeld†
* Delhi Technological University
New Delhi, Delhi 110042 India
† Robotics Institute, Carnegie Mellon University
Pittsburgh PA 15213 USA
Email: * suryansh@red-itech.com, † bmin@cs.cmu.edu, mbdias@ri.cmu.edu, steinfeld@cmu.edu

***Abstract*—This paper focuses on the system development for an android app for blind and visually impaired users for outdoor navigation needs. The application is called NavPal Outdoor app. The larger aim of the app is to support blind and visually impaired user to navigate outdoors independently without help from any other person or device. The application utilizes the hardware of an android supported smartphone to get accurate and realtime information about the route and navigation information. The app facilitates the users as well as other sighted or visually impaired user to act as trusted sources to add dynamic information to the map for updating the users with accurate and realtime information about routes. The paper highlights the main architecture of the application, different components of the app and their respective functionality.**

***Index Terms*—Android application, Breadcrumb approach, Contact destination principle, Geo coding, Haptic and speech interface, NavPal[1],Navigation Aid,Smart-phone app, Trusted sources, Trusted user interface**

I. Introduction

The NavPal Outdoor app is an android application which is designed to facilitate the outdoor navigation needs of blind and visually impaired user. The app is a standalone unit which uses the different hardware capabilities of an android supported smartphone along with communicating with several servers to return accurate realtime route and navigation information. Therefore facilitating the users to navigate independently and safely in an outdoor environment. The app calibrates the routes according to the user's preference and gives essential information about route and path, to help the user localize, navigate and create a log for future travel. Through this work, we will be exploring the specific challenges encountered by blind and visually impaired travelers and their counter measures when navigating in an outdoor environment.

A. Motivation

There are more than 15 million people in the United States of America who are blind or visually impaired, and majority of this population face some critical challenges when navigating outdoors. Safety and independence of navigation are some of the most cardinal issues to be addressed. There is a huge risk involved for blind and visually impaired users to navigate independently outdoors due to limited nature of information present for the route ahead. For example its very difficult for blind and visually impaired users to estimate pot-holes and ditches in a route and hence safety becomes an issue. Also day-to-day activities such as using transit systems remain challenging tasks for people with visual impairments even though the use of transit systems is often a key factor for participation in employment, educational, social, and cultural opportunities.

The current tools and devices that exist in today's market space that help blind and visually impaired travelers navigate outdoors are limited. Some of them are very expensive and not accessible to a larger community. Also many of these tools depend upon external hardware which limits their usage and often needs to be replaced at regular intervals to avoid faulty measurements. Some tools use the infrastructure provided by the cites or states like bluetooth beacon device, traffic light alarms etc which may not be present universally in each city thus making them limited. The breadth of information about the route and navigation instruction provided by these devices are also limited. Hence the overall accessibility of these devices or tools are limited to a much smaller population. Even today a majority of the target users just rely on basic methods like using white-cane for finding navigable path, using dog guides for finding routes etc. Also the users often require the help of other individuals for their daily navigation needs. It may be noted that mobility of the blind travelers is also limited in cases where clear navigation information is not present. For example a case where the user is unable to find benches in parks where he/she usually sit. All these problems are widely common within the blind and visually impaired travelers community and must be addressed.

To address these problems NavPal Outdoor app project was initiated to conceptualize an application on an android smartphone platform. The main motivation to develop NavPal Outdoor app over an android smartphone device was due

[1] ©TechBridgeWorld
NavPal project was initiated by TechBridgeWorld research group along with the Robotics Institute at Carnegie Mellon University. The first author was affiliated with Robotics Institute Carnegie Mellon University TechBridgeWorld research group for the duration of this research. For more information on the NavPal Project please visit http://www.cs.cmu.edu/ navpal/index.html.

to the fact that today smartphone technology has become high accessible to a very large community and it provides a platform to integrate state-of-the-art technology for outdoor navigation. These devices have become more reliable with higher processing power and loads of sensor attached as peripheral systems to the device, all together at a very cheap cost. With the advancement in the voice recognition, text-to-speech and haptic technology it has also become easier for the blind and visually impaired users to access, communicate and work on the smartphone device, making them a promising and independent solution for developing apps for blind and visually impaired travelers. Along with the development of the smartphone technology, android operating system has also become highly popular. The ability to implement cheap, easy and highly customize-able apps have made android an excellent platform to develop new tools for navigation. Traditionally blind and visually impaired users preferred iPhone device with IOS operating system rather than android technology, as iPhone devices initially offered more user friendly environment and feature for the blind users. However with rapid development in assistive android technology, many blind and visually impaired users have switched over to android devices making android a platform for sustainable development.

B. Approaches

The NavPal Outdoor app project aims at creating a standalone app on an android supported smartphone platform to help assist blind and visually impaired users navigate independently and safely. One of the major aims of NavPal Outdoor app is to be highly accessible to the blind and visually impaired users so that they can easily communicate with the app and effectively customize the app as per their preference. The NavPal Outdoor app via android device harness the power to find locations of the user while navigating using GPS and network services and reduce the error of the location finding services using the INS or inertial measurement sensors like accelerometer, gyrometer and compass by tracking the last few location positions of the user and correspondingly maintaining the accuracy of the GPS data.

The app parse information about the path and navigation instructions from multiple servers like Google, Wiki and local servers using the cellular network capabilities (2G/3G data) and also allows users as well as other secondary sighted and blind or visually impaired users to contribute to the route or navigation information. This feature introduces two important domain of the application that is Trusted Sources contribution and Breadcrumb generation. The data from these sources are dynamically represented and updated on the map as way-points, in order to maintain the navigation data as accurate as possible with regard to the changing environment. The app communicates the instructions and result to the users via text-to-speech interface and vibrations, while taking feedback from the users via accessible gesture drawing and voice recognition applications. The gesture can be customized as per the preference of the user for higher accessible communication with the app.

These approaches allow the users to get realtime and accurate information about the route and navigation instructions even in dynamically changing environments, with an ease of using the app as per their preference. The way NavPal Outdoor app communicates with the user is such that it tries to sync the navigation instructions along with the inherent capability of the user to navigate independently, so that the users can utilize their navigation training while getting essentially navigation information from NavPal Outdoor app.

C. Objective

The main objective of this paper is to highlight the primary architecture of NavPal Outdoor app using an android platform, its different components and their respective functionality. The study of the architecture and design will help the readers to understand how NavPal Outdoor app works and its ability and efficacy to act as a standalone unit for facilitating outdoor navigation for blind and visually impaired travelers. To develop upon the understanding of NavPal outdoor app, the paper has been organized as follows. In section II,we discuss the related work. Section III describes the different components of the NavPal outdoor app and their respective functionality to achieve outdoor navigation. We validate the performance of the app with respect to its accessibility and ease of use in the sections IV where we underline an experiment conducted with a focus group to test NavPal Outdoor app's performance with the target users. We highlight the results from the user testing and finally conclude in section V.

II. RELATED WORK

The fundamental tools for navigation available for blind and visually impaired users include white canes and dog guides. A majority of blind and visually impaired users prefer to use white canes to understand the obstacle and path ahead of them, while the dog guides personified,are like their GPS navigation medium. However, even though these tools and methods are conventional and fundamental, they do not provide the accurate and dynamic information about the route and the path and are prone to navigational errors. These tools and methods alone cannot facilitate for independent and safe navigation.Some of the other navigation tools for the target users include guidance robots. These robots help users localize and navigate in indoor and outdoor environments. The robots are packed with sensors like inertial measurement sensors, GPS, WIFI receivers, laser scanner etc, that help the bot analyses the environment and plan a route strategy. The robots instructs and guides the user via touch or force and voice interactions. Major limitations to these robot based guidance platforms is that, the robot guides are very expensive and hence not accessible to a large community. Also the fact that the robots locomotion is restricted by its own dynamics and degree of freedom brings the limitation of the robot to navigate in different environments and paths.

Fig. 1. Blind traveler using conventional methods to navigate; white cane, guide dog

Fig. 2. Robot quadruped guiding a blind traveler

Other tools include GPS based navigation device that guide the users with a distance based navigation routine. These devices rely on the pre-fed data about the environment and simultaneous GPS location. *Trekker Breeze* is an popular GPS hand-held tool which comes under the same category. Some of these devices communicate the instructions to the users via voice interface and take input form them via fixed logic buttons present on the device. Even thou popular in today's market space, many of these hand-held GPS devices only act as route and surrounding information sources and hence are not efficacious for independent and safe navigation. Other form of these GPS hand-held device are smartphones which are far more complicated and accurate for outdoor navigation for blind and visually impaired users. Today with the advancements in the smartphone based technologies these device have now become the modern tool for navigation for the blind and visually impaired community. The main advantage of the smartphone devices is that they are encompassing devices with many essential sensors for outdoor navigation along with ability to server as a effective programmatic platform for developing mobile apps. Many developers around the world are working in the area of developing navigation tools for both indoor and outdoor environment for blind and visually impaired users. TechBridgeWorld's NavPal Indoor app project is an excellent example that help blind and visually impaired user navigate in an indoor environment. NavPal Indoor project was focused on generating or parsing indoor maps of buildings and analysis them to draw paths and navigation instructions. Some smartphone projects for blind and visually impaired users for outdoor navigation include Sendero group's GPS

Fig. 3. NavPal indoor application

enabled iPhone app. The app helps user localize and provides navigation instructions on an Iphone device. The navigation information is provided to the user for various intersections or at system defined way-points. The limitation to the Sendero's app is that is does not provide realtime information about the environment or path changes which are crucial to the target users. Other representative smartphone GPS apps targeted toward visually impaired users include BlindSquare and Ariadne GPS. BlindSquare is a very popular app running on the iOS platform. It uses points of interest information from Open Street Map and allows navigation to the point of the interest. Loadstone GPS is GPS navigation software for visually impaired users. This software runs on Nokia mobile phones and uses external devices such as screen readers and a Bluetooth GPS receiver. The primary challenges for such commercial navigation solutions come from their use of their own map modules to provide map information. Because of this, visually impaired users need to pay to run those apps on their smartphones and require continuous updates on a map that incurs additional charges. Moreover, most route planning apps are inflexible to deal with dynamic changes, for example handling a case of path blockage due to construction, and are not designed to be updated and refined by users.The notion of breadcrumbs and trusted source contributions via the NavPal Outdoor app address these issues in much details in section III.

III. METHODOLOGY

The main aim of *NavPal Outdoor Application* is to develop an open source platform which is highly accessible to the blind and visually impaired communities and facilitates for their outdoor navigation needs, along with ensuring the safety and independence of navigation. The platform acts as an standalone unit for providing navigation data with accuracy and updating dynamic realtime information about the navigation instructions and routes to the user. This paper aims to highlight the main system architecture of the NavPal Outdoor app, along with describing the various sub-modules, their functionality and their respective contributions to help facilitate outdoor navigation.

A. Prerequisites for NavPal Outdoor app operation

The main architecture of the app requires the users to calibrate their respective android devices before usage. Thus

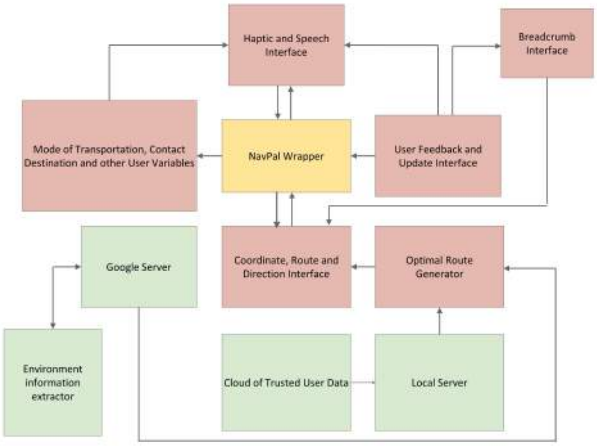
certain prerequisites must be followed in order for the app to perform normally. The prerequisites have been cites as follows:

1) The smartphone device must have a inbuilt GPS receiver, accelerometer and compass sensors.
2) The device must also have an active 3G/2G/Edge cellular data connectivity.
3) The device must have destinations stored in form of contacts with atleast a contact name and contact address.

B. Main Architecture

The main architecture of NavPal Outdoor app has been designed and written over an android framework with a Linux-3 file structure operating system. Fig.4 highlights the main architecture of the NavPal outdoor application. The *NavPal wrapper* in the Fig.4 is the main the user

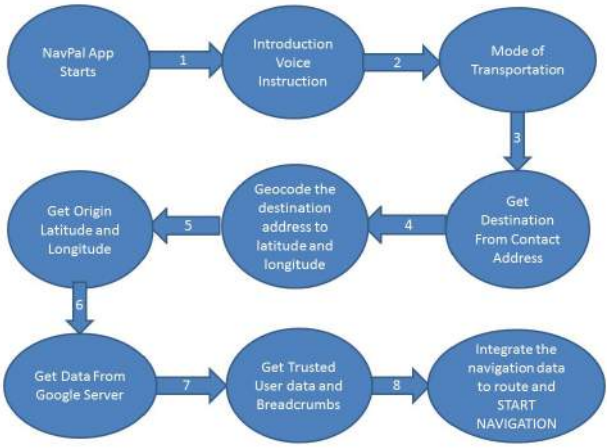
Fig. 4. NavPal outdoor application architecture



interaction element of the app. The wrapper communicates with the various sub-modules to fetch different classes of information needed at different intervals to facilitate the outdoor navigation. The wrapper also facilitates and maintains the flow of the control within the application code structure. The users is able to interact with the wrapper using the Haptic and Speech Interface, reference section III-C, which provides instructions and system quires to the user via a audio text-to-speech interface and vibrations while the user provides his/her respective feedback and preference control parameters via accessible gesture drawings or via speech recognition modules. The wrapper is also responsible for communicating with various servers in order to return navigation and route information. The wrapper is customize-able as per the users preference, with respect to system verbosity, speech rate and changing accessible gesture list which are discussed in section III-C. The NavPal wrapper is also responsible for pinging and acquiring data from accelerometer sensor and compass,inbuilt in the smartphone to help the user localize and get additional orientation information for navigation instructions. The data from these sensors are also used explicitly when GPS signals are lost during navigation, to maintain an probabilistic estimation of the users current location. The other sub-modules of NavPal Outdoor app's architecture like the breadcrumb interface and the trusted user interface are the modules that help integrate dynamic environment and route information to the navigation list. These modules generate additional way-points on the navigation path that would help the user localize and navigate safely in an familiar or unfamiliar and changing environment. The additional way-point data via these interfaces are contributed either by the primary user or via secondary sighted or visually impaired users, based on their navigation experience and knowledge about the environment. The other sub-modules like *Mode of Transportation, Contact Destination and other User Variables* are used for calibrating the app with respect to users navigation variables like intended destination, which is parsed using the Contact Destination Principle, and user's preferred mode of transportation. The component *Coordinate, Route and Direction Interface* and *Optimal Route Generator*, are responsible for fetching the navigation and route instructions and then mapping the optimal route by integrating trusted source data, breadcrumb data and Google server data for navigation. The traffic on the possible routes is also analyzed, and the best possible route is selected for the user based on selected transport medium.

The flow of the code structure has been elaborated in the Fig.5. The flow shown in the figure illustrates the

Fig. 5. Flow of control structure in NavPal Outdoor App



sequence of the calling of different sub-modules during the initialization of the app and before the navigation. During the initialization stage of the app, the users are required to selected mode of transportation, verbosity and contact destination. Upon getting the desired parameters, the system coverts the destination and origin locations into geo-coordinates and pings the various servers like Google server, Wiki server and Local server to plot an initial route. This route is augmented with the date from selected trusted sources and breadcrumbs and a final optimal route is generated. Upon starting the navigation, the user is given route and navigation information at different way-point intervals as shown in

Fig. 6. Snapshot of NavPal Outdoor App functioning

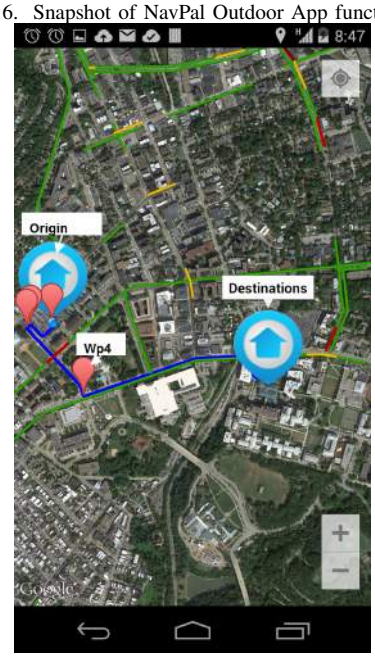


Fig.6. The navigation information is reiterated to the user at various instances of approaching the next way-point. At intersections the street name and the direction of travel are reiterated as well as the nearby points of interest if any, are mentioned to the user while navigation for better localization and orientation. The different sub-modules and their detailed functioning is mentioned in the sub-sections below.

C. Haptic and Speech Interface

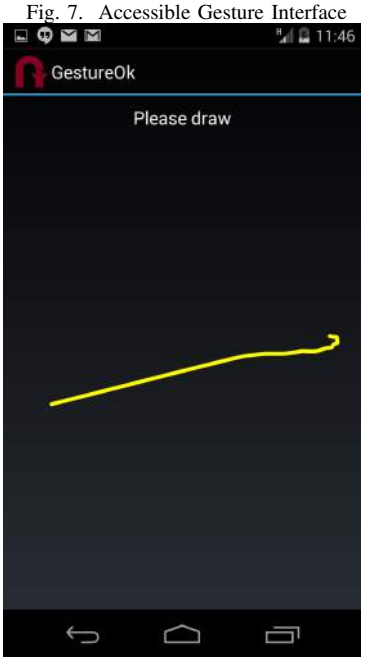
The Haptic and Speech interface establishes communication between the user and the app. The Haptic and Speech interface is made of the 4 sub-components, namely a)Text-to-Speech interface, b) Speech Recognition interface, c) Vibration interface and d) Accessible gesture interface. The Text-to-Speech interface helps read out important instruction to the user. The app verbally communicates with the user via this interface. The speed of speech is configurable as per the users preference.Next the speech recognition interface, which is made up of the Google speech recognition API, aims to understand what the user is trying to indicate. The Speech recognition api, in the NavPal Outdoor app has been set to *Action-Web-Search* classifier so as to increase the probability of identifying a larger set of keywords within the short instruction user's speech. The app gives users certain keyword to speak from, which are better estimated by the Action-Web-Search classifier. Further the speech recognition API's result is evaluated with respect to the keyword via*Levenshtein distance* algorithm. Mathematically Levenshtein distance between two strings a, b is given by below: Note that the first element in the minimum corresponds to deletion (from a to b), the second to insertion and the third to match or mismatch, depending on whether the respective symbols are the same. The higher probable results left, are

$$\mathrm{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \mathrm{lev}_{a,b}(i-1,j) + 1 \\ \mathrm{lev}_{a,b}(i,j-1) + 1 \\ \mathrm{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

spoken to the user for further accurate selection.The terms i and j represent the index of the character in a string a and b.

Vibration interface is used to indicates the users when

Fig. 7. Accessible Gesture Interface



something illegal with respect to control structure has happened like pressing the home button during run-time or bad attempts while inputting the data. Finally the device vibrates when the user takes a discourse from the actual path during navigation. Finally Accessible gesture interface gives the user the capability to answer YES-NO question or navigate to the setting manger for accessing the preferences of the user, wherein the users can change verbosity levels and speech speed, or to activate different route an navigation update sub-modules like the breadcrumb and trusted user interface while navigating. The different set of gesture can customized as per users preference. An example of accessible gesture is shown in the Fig.7, where the user draws from left to right, which the system interprets as a positive response.

D. Selecting the Transport and Contact Destination Principle

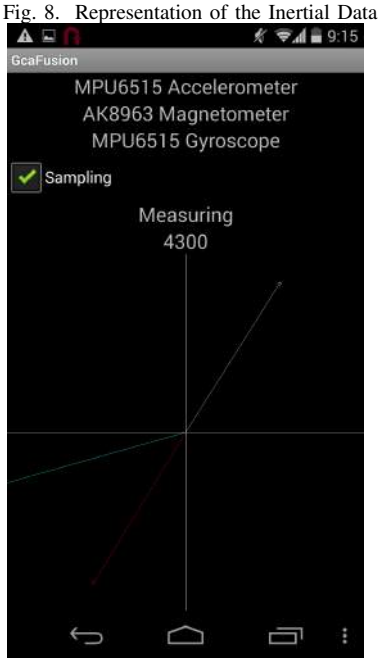
NavPal Outdoor app allows users to select from three mode of transportation, namely a)Walking, b) Taking a Bus and c) Taking a car/cab. The users are able to select the modes of transport via the accessible gesture interface shown in the Fig.7. One by one each option is spoken to the user and correspondingly the user gives his/her input via accessible gesture. Ones the transport type is set the NavPal wrapper shifts the control to the contact destination interface. *Contact destination principle* is a method in which the user

must store the possible or intended destinations in the contact fields along with their respective address. Ones this component is initiated, it calls in the speech recognition module, reference section III-C, and instructs the users to speak the desired contact name for extracting the destination address. Upon analyzing the speech of the user the system compares the best possible results with respect to the contact list and correspondingly returns a list of possible, higher probable, destinations, here contact name. Now this list is spoken to the user, and correspondingly the user indicates its choice via accessible gesture. As soon as the first positive result is acquired the module parses the contact destination from the list and stores it under the destination variable.

E. Localization, Routing and Path Extractor

Ones the destination has been set and the user is all set for navigation, the Localization, Routing and Path Extractor sub-modules of the NavPal Outdoor app become active. The aim of the Localization interface is to find user's current location. The localizer pings the GPS receiver and the Network manager to update the device with best possible location estimation. Coordinates received have some inherent noise and thus filtering of the user's location is required. The filtering of the data is done with help from the inertial sensors after an initial position estimation. The representation of the inertial navigation data generated by the sensor is given by the Fig.8 where the data from each sensor is marked in the X-Y domain. To reduce the noise, the GPS receiver and Network

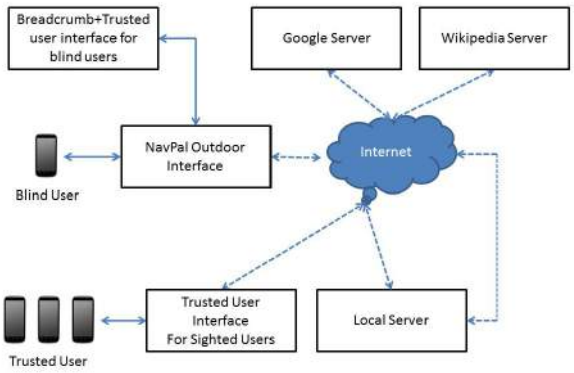
Fig. 8. Representation of the Inertial Data



manager is pinged several times for current user location estimation. It may be noted that the noise from GPS is lesser than that from the Network manager. As the data received are being stacked the accuracy of the data is being monitored with respect to the previous data estimation. A pool of such data is collected and averaged weighted sum of the accuracy is taken to filter the coordinates received. Upon reaching an acceptable accuracy, the coordinates got are set as origin variable. Now the next step is to convert the destination address into geo-coordinates. Thus sub-module *Geo Coder interface* is launched and it pings the Google server with the address of the destination and correspondingly receive the geo-coordinates (latitude and longitude) of the destination. Now the app is ready to extract the route and navigation instructions.

Thus the Routing and Path Extractor sub-module comes into plays. This module first pings the Google server via HTTP request to return a number of routes from origin to destination. The Google service accepts the HTTP request along with the geo-coordinates of origin and destination, compass heading and mode of transportation data. Upon processing the request the Google server send a JSON object over the a HTTP request back to the NavPal Outdoor application. The JSON object contains a array of routes, possible set instructions and way-point coordinates. Upon receiving the JSON object, the NavPal wrapper executes the *JSON parser interface*. This interface decodes the JSON data received into list of way-points, navigation instruction and routes information. The Path Extractor interface also refer additional server like Wiki for getting information about the surrounding and adjacent street names, for better localizing and orienting the user with direction. Now a

Fig. 9. Working of the Route and Path Extractor interface



route is prepared and this route is augmented with the data from previously stored breadcrumbs and trusted sources contribution. The Path Extractor interface first pings the breadcrumb interface for any previously recorded data. If there is any previous records, those breadcrumbs are returned in the form of way-points onto the map. It must be noted that the breadcrumb data is unique to each user.

After breadcrumb interface the Trusted user interface is called in by the Path Extractor sub-module. The trusted user interface looks for the possible zip-code regions, where the

user will be navigating by pinging the zip-codes of the origin and the destination location. All the data associated with these zip-codes contributed by the selected trusted sources, from the trusted user interface is returned to the Path Extractor.The Fig.9 shows the above process in general. Ones all the data from the different contributing interfaces are parsed, the app starts to make an optimal final route from all the data obtained along with the different navigation instructions associated. With this the Route and Path Extractor interface transfers

Fig. 10. Route and Navigation Instruction Mapped

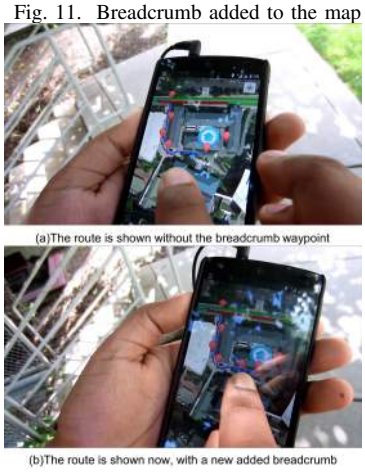
the control flow of the application back to NavPal wrapper. The example of the working of the Route and Path extractor mapping information has been shown in the Fig.10 where the information about the route is shown by the underlying navigation instructions.

F. Breadcrumb Interface

The notion of breadcrumbs is one of the major concepts of the NavPal outdoor application that help users add additional information to the navigation and route for familiar paths. There are many keynote instruction or orientation tips that the users develop while frequently traveling over the same route. The breadcrumb interface gives user the ability to add new way-points data as voice memos for each respective data points, so that when they travel next through the vicinity of these way-points, the recorded data will be played back to the user. This gives user the ability to add additional information in the route by recording critical localization and orientation information as per the user's experience.

The breadcrumb interface has two different calls in the NavPal Outdoor app's architecture. The first call is made by the system, during the Path Extractor interface [reference Section III-E]. Here the breadcrumb interface, forwards all the prerecorded data to the path extractor interface for planning of the route. The Second call is made by the user, for adding new breadcrumb data points. This call is made during the navigation stage, and the user can call the breadcrumb interface multiple times. When the breadcrumb interface is called, the user's most current position is recorded as the way-point or breadcrumb. Upon launching the interface the app instructs the user to record their message, for the particular breadcrumb. After the recording is complete the system plays back the recorded message to the user, and gives the option to record again if the user is not satisfied. Upon user's approval, then the audio recorded is passed to the data point along with the way-point and the compass location with respect to the way-point added, which is stored in a shared preference android data structure. After the addition of the new way-point the breadcrumb interface transfers the control back to the NavPal wrapper along with a new breadcrumb data added. Fig.11 shows an example of adding

Fig. 11. Breadcrumb added to the map


(a)The route is shown without the breadcrumb waypoint

(b)The route is shown now, with a new added breadcrumb

the breadcrumb data to the map. The NavPal wrapper adds that way-point immediately on the map with the recorded data as an pointer instruction to that breadcrumb.

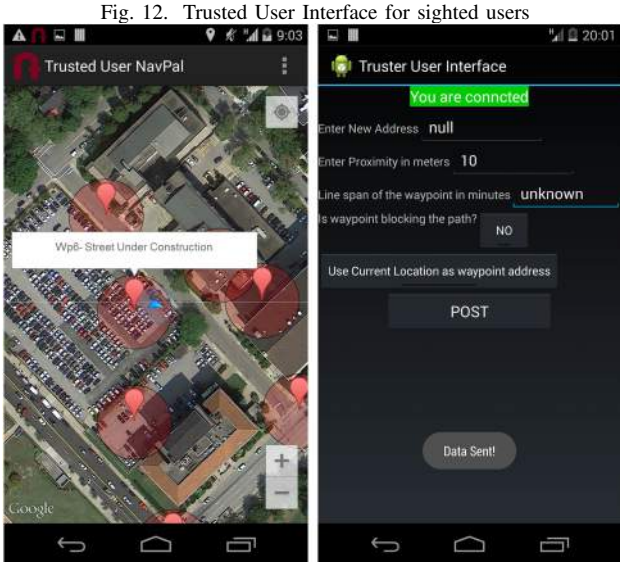
G. Trusted User Interface

The Trusted User interface is one of the core concepts for adding realtime data about the environment and the path to the user's navigation map. Trusted user interface facilitates the users to stay updated about the changes happening in their route and corresponding effects it has on their navigation. With the trusted user interface other or secondary sighted or visually impaired users can record their observation about the change in the environment and route as per their experience. This data can be utilized by the primary blind or visually impaired traveler when navigating on the same route. For example, if a street is under renovation, and if a trusted source traveling via this route notice this change then he/she can add this data for the use of blind and visually impaired travelers. The data added by the traveler can be utilized by the NavPal Outdoor app to effectively route the blind and visually impaired users such that they do not encounter this street

under renovation or if there exist no parallel route, atleast the users can be alarmed about the scenario when reaching in the proximity of this street. This approach helps the user to have an additional realtime information about the environment and the routes.The addition of data and its access via the trusted user interface is maintained and controlled on an external server maintained by TechBridgeWorld organization. The data contributed has certain attributes, that must be specified by the users during their respective contributions. These attributes help the Path Extractor and the NavPal wrapper understand the dynamics of the contributed data. The main attributes for the data points are, a) Is way-point a blocking obstacle, second any text/audio associated with the data point explaining its dynamics. Third the proximity associated with the data point, fourth the life-span of this data-point.Along with these attributes the system also attaches few attributes to the data set, that are,one the time-stamp or the time of origin of this data point and two the identification of the trusted source via referencing the cell-phone number.

The trusted user interface is integrated within the NavPal application for the blind and visually impaired user for contributing as trusted sources. The users can access the trusted user interface in the NavPal application via accessible gestures. Ones the trusted user interface starts, the users are given two options, one to use current position or two use a new location to contribute. The blind and visually impaired users input the value of each attribute via accessible gestures and audio recording interfaces. The users can define a life time in hours and minutes or can leave this attribute as an unknown life-time value. The app compiles all this data into a JSON object and includes a time-stamp of origin and an identify (Cell phone number) of the trusted source. This JSON data is sent and stored over the external server. A similar but

Fig. 12. Trusted User Interface for sighted users



independent (of NavPal Outdoor app) interface is available for the sighted user to contribute data as trusted sources. The Fig.12 shows the trusted user interface for the sighted user wherein the users have a full Google map environment and they can navigate to different places via entering the address or scrolling to the data point address. Ones the sighted user have identified the data-point, they can simply tap near that point on the map and a way-point will appear. After the way-point has been generated the sighted users will be asked to provide the values to the different attributes of the data point. A JSON object is created ones all the attributes values have been set and the resultant JSON is stored over the external server via an HTTP link between the app and the server. The server stores this data point associated with the user's identification number in a form of a table. This is how the data from a trusted sources is stored onto the server.The trusted sources data is retrieved by the NavPal Outdoor app for those trusted sources that have been selected by the user. The interface allows the users to select a list of people as trusted sources from their contact list. This list is stored in the android's internal memory and can be edited by going in the setting manager interface. The data from each source is called first during the NavPal Outdoor app's initializing, that is when the Path Extractor interface starts the trusted sources data is queried via a HTTP request to the external server for list of trusted sources selected and data required for desired zip-code extensions. The resultant data is sent as JSON object back to NavPal Outdoor app which is further, decoded to add new way-points and their respective information on the map.

H. Re-routing and Path Divergence

The NavPal Outdoor app also accounts for the cases of path divergence. When the user diverges from the path then the NavPal Outdoor app warns the user about the discourse and repeats corrected navigation instruction, such that, if followed, gets the user back to the initial course. The amount of deviation is predicted by estimating the distance between user's current location to next desired way-point and the distance between the user's current location to the last visited way-point. The distance is subtracted in order to find the magnitude of the deviation. After a certain magnitude of deviation the app instructs the user to stop, and the app re-calibrates the route with respect to the user's current location and to the desired destination.

IV. EXPERIMENT AND USER TESTING

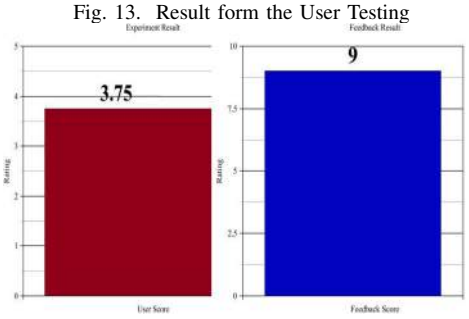
The NavPal Outdoor app was tested in Pittsburgh USA, by the TechBridgeWorld team to evaluate the performance of the app in a realtime outdoor navigation scenario. The aim of the experiment was to test various components of the NavPal Outdoor app and evaluate how they performed in different situations while navigating. The experiment was conducted by physically navigating with the help of the NavPal Outdoor app from the Robotics Institute, Carnegie Mellon University to Carnegie Natural History Museum. During the navigation the team tested various components of the NavPal Outdoor app, like adding breadcrumbs, checking the speech recognition and haptic interface, checking the apps performance while

deviating from the route and finally testing the navigation instructions provided by the application. The destination Carnegie Museum of Natural History was stored as contact name Natural History Museum in the team's android device. As the system started the navigation, it instructed the team firstly about the duration and distance to the destination and subsequently instructed the user with the first navigation instruction. As the team follows the navigation instructions the system waits for the user to approach the next way-point and then subsequently instructs the user about the next navigation instruction. To test the concept of breadcrumbs the team halted at a spot and starts the breadcrumb interface. The breadcrumb interface started and asked the user to record the message for this breadcrumb. The team recorded a message and upon a short tap on the screen the user finalized the message and a new way-point wad added to the map. This demonstrated the concept of adding breadcrumb in the map.

While navigating the team intentionally diverged near the destination way-point to test for the app performance in such scenarios. The system reacted immediately and informed the team about the discourse and the wrong direction warning was played to the user, with instruction to return to the previous course. The team continued to move in the wrong direction and in reaction, the app started to re-calibrate a new route and navigation instructions to help the team reach the desired destination. Upon reaching the destination, the team retracted its path back to the starting point. During the return the team re-encountered the breadcrumb stored previously and as the team approached the breadcrumb, the recorded data associated with that breadcrumb was played back to the team during navigation. Thus exhibiting the concept behind the breadcrumbs interface.

The NavPal outdoor app was also tested with a focused group comprising of blind and visually impaired user who volunteered to test the accessibility and ease of use of the NavPal Outdoor app. A total of 4 users visited Newell-Simon Hall on 25th July 2014. During the testing the participants were asked to set-up the NavPal Outdoor app, with respect to 1) adding a contact, 2) open the app, 3) selecting mode of transport,4) selected a contact name via speech recognition interface and 5) setting up the data on the map to start the navigation. The users performance was rated by the system, where each positive step towards a certain desired flow was marked +1 while a wrong step was marked as -1. Finally the users rated the application according to their experience and gave their respective feedback. The result of the user testing came out to be positive. The users as well as the internal system highly rated the apps performance and the corresponding user ability to work on the app. The Fig.13 shows the aggregated results from the user testing, where the system that is the apps internal marker rated the users on a scale of 0 to 5 in the Red color graph while the users rated the app's performance with respect to accessibility and ease of use on the scale of 0 to 10 in the Blue colored graph. The

Fig. 13. Result form the User Testing



results show that the app was found highly accessible and ease to use by the focus group.

V. CONCLUSION

In conclusion the Navpal Outdoor app's basic architecture was developed and tested with blind and visually impaired users for accessibility. The app's performance with respect to the blind and visually impaired user testing, showed that the app was highly accessible and easy to use. Some of the future work for NavPal Outdoor app, is to remove the dependence on the Google servers, and to make and plan custom routes based on user's preference. Other future works would include integration of the app with robots to facilitate the users to have higher accessibility and navigation capabilities.

VI. REFERENCES

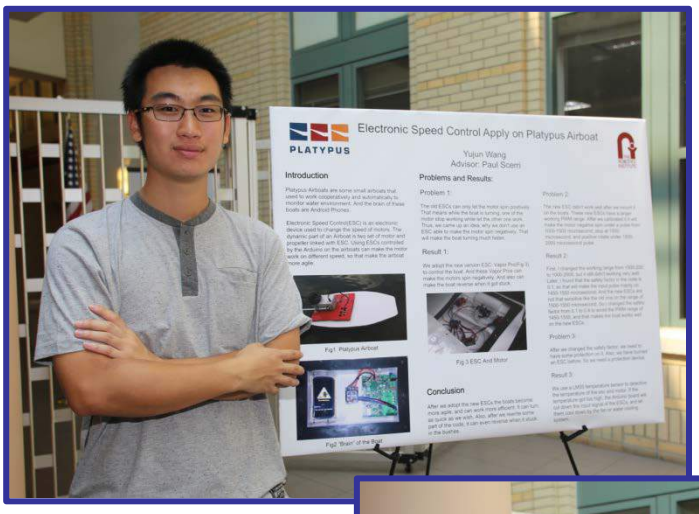
1) Hend Gedawy,"Designing interface for indoor navigation system for visually Impaired", M.S Thesis, Dept. CS CMU, PA, 2011.
2) Kannan, Balajee, et al. "Localization, Route Planning, and Smartphone Interface for Indoor Navigation", Co-operative Robots and Sensor Networks. Springer Berlin Heidelberg, Volume 507, pp 39-59, 2014.
3) N. Kothari, B. Kannan, E. D. Glasgwow, and M. B. Dias, "Robust Indoor Localization on a Commercial Smart Phone", Procedia Computer Science, vol. 10, pp. 1114-1120, 2012.
4) J. A. Hesch and S. I. Roumeliotis, "An Indoor Localization Aid for the Visually Impaired ", in 2007 IEEE International Conference on Robotics and Automation, 2007, pp. 3545-3551.
5) P. Narasimhan, R. Gandhi, and D. Rossi,"Smartphone-based Assistive Technologies for the Blind", in Proceedings of the 2009 International Conference on Compilers, Architecture, and Synthesis for Embedded Systems, New York, NY, USA, 2009, pp. 223-232.

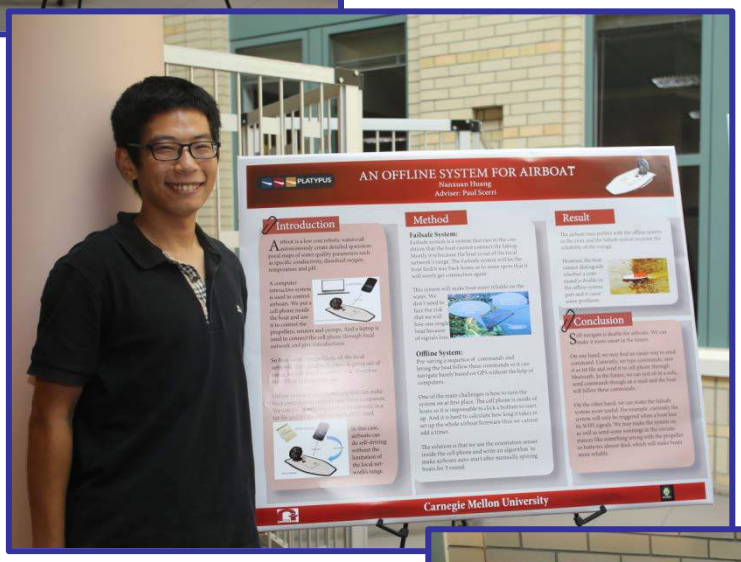
VII. ACKNOWLEDGMENT

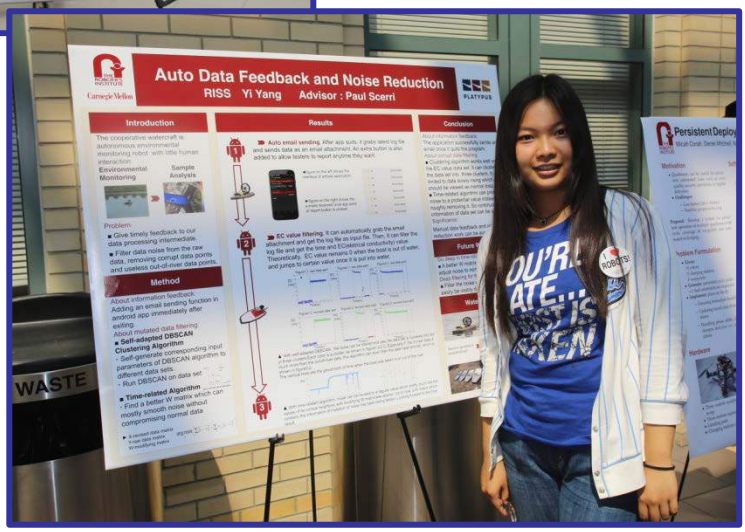
This work was funded by a grant (H133E080019) from the United States Department of Education through the National Institute on Disability and Rehabilitation Research and a grant (National Robotics Initiative, IIS-1317989) from the National Science Foundation. The authors also thank the participants in the user studies that informed this work, and Ermine Teves who assisted with different aspects of this work.

Yujun Wang, Nanxuan Huang, and Yi Yang

RISS 2014







Platypus Cooperate Robotic Watercraft Platform

Electronic Speed Control, Offline System, Failsafe and Auto-data Feedback

Yujun Wang
Robotics Institute Summer Scholar
Carnegie Mellon University
Pittsburgh, U.S.
Geoffreywang1990@gmail.com

Nanxuan Huang
Robotics Institute Summer Scholar
Carnegie Mellon University
Pittsburgh, U.S.
Hnx00001@gmail.com

Yi Yang
Robotics Institute Summer Scholar
Carnegie Mellon University
Pittsburgh, U.S.
elaineyang127@gmail.com

***Abstract*—this paper addresses the application of Electronic Speed Control (ESC) Vapor Pro using a Platypus Cooperative Robotic Watercraft (CRW), and a robotic algorithm designed for automatic control used on the watercraft. An offline system will enable the airboat to self-navigate without any extra help. An Android App was developed based on this algorithm in order to transfer users order into mechanical order thereby controlling the boat's hardware directly. A failsafe system was also designed to make sure the boat would come back home in unexpected circumstances. An automatic data feedback system was designed to automatically grab the log file and send an email with attachment to the users. These algorithms will make the airboat more reliable.**

Keywords—platypus; CRW; ESC; Vapor Pro; offline; failsafe system; automatic navigation; automatic data feedback;

I. INTRODUCTION

The Platypus Cooperative Robotic Watercraft (CRW) platform is made of several small robotic boats. These vessels are used to monitor water environment automatically and cooperatively. The goal is to build a fleet of boats that work together to monitor a large area of an aquatic environment. In order to have lots of these boat, we need to keep each of the boats low cost.

The brains of these boats are Android phones. The boats use the GPS and WIFI of the phone for communication with other boats and the computer. Also, with the help of the low power sensors and processor on the phone, the boat can work very efficiently.

Here are two generations of the watercraft:

Lutra 1 series are using a fan placed above the water surface as propulsion, which make the boat run in low speed, but without interference from the aquatic plant.

Lutra 2 series are driven by two motors with propellers below the water. Electronic Speed Control (ESC) is an electronic circuit used to change the speed of an electric motor. Using ESCs on the airboat enables the motor to work at different speeds. This enables the vessels to turn quickly and waste less energy while working. So the boats can work more efficient than Lutra 1 series. But the disadvantage of this generation is that the propeller can get stuck on waterweeds. Thus we still employ the Lutra 1 in certain environments.

The problem we were facing with the Lutra 2 series is that sometimes the boat will get stuck in the bush in the river, and the old version ESC can only make the motor rotate positively. We need a new kind of ESC that can make the watercraft reverse. The Vapor Pro is a waterproof ESC with a fan cooling system that can make the motors spin bidirectional, which meets our desire. After adopting the new ESCs in the Lutra 2 series, the boats turn more quickly and are even able to reverse after going aground.

This paper describes the work we have done to make the ESC Vapor Pro work on the boat, and new functions: Offline System and Failsafe System.

II. ROBOTIC BOATS

A. Structure of a Boat

The boat (Fig. 1) is made of ABS plastic. ABS plastic was chosen because of its low cost. There is a cap with a rubber waterproof seal for the cabin to keep the water away from the electronic devices in the cabin.

In the cabin (Fig. 2), we place the crucial part of the boat inside: the brain (an Android phone), a battery, an Arduino board, an Electronic board, a pump, two ESCs and two motors. All of these things are linked by wire and cable.

There are two propellers attached to the two motors placed under the boat for propulsion.

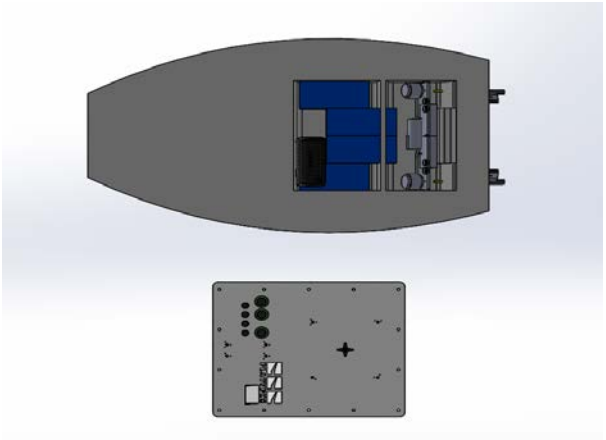


Fig. 1 3D Model of Lutra 2.0

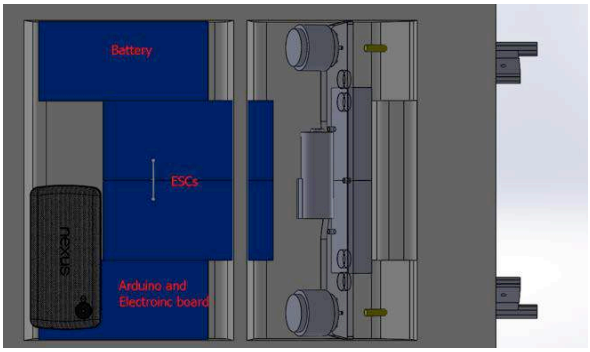


Fig. 2 3D Model of the Cabin in Lutra 2.0

B. *Software Structure*

One of the design considerations is to make the boats able to be operated by people who are not experts in robotics [1]. Therefore, they must be made as simple as possible. All that is needed to operate a boat is a laptop or tablet, an Android Phone, and the boat (Fig. 3).

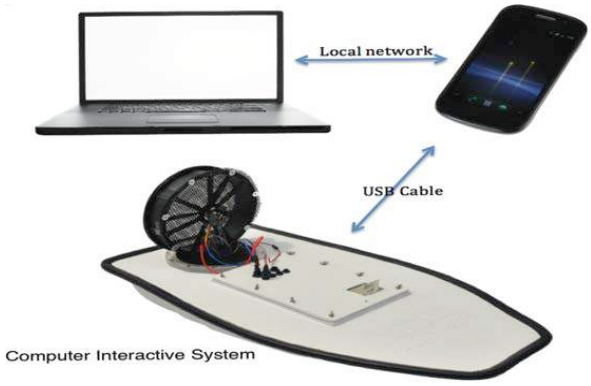


Fig. 3 the Original Interactive Mechanism

We need to link the laptop or tablet and the Android to the same Local Area Network. Then we assign an area that the boats need to test by computer or tablet. At the meantime, the Android phone on a boat will get the waypoints it needs to go to from the laptop or tablet. With these points, the Android phone will calculate the speed and the direction the boat needs to reach each point by Proportion Integration Differentiation (PID) control method, and then send Pulse Width Modulation (PWM) signal to the Arduino board. Meanwhile the Arduino board will translate the signal to the ESCs and make the motor work.

To communicate with the phone, sensors, ESCs, and motors, we used Arduino due to control the external sensors and motor (Fig. 4). The Arduino board gets the data from the phone via USB cable. The ESCs are directly connected to the Arduino and controlled by the board. Depending on the sensor, external sensors are all plugged directly into the Arduino I/O port.

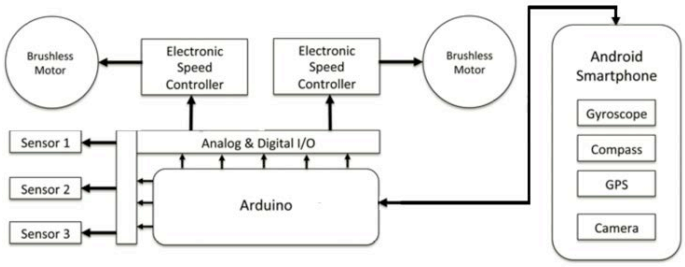


Fig. 4 System Architecture Diagram

III. Electronic Speed Control

A. *How to Calibrate the Vapor Pro ESC*

In order to make the boat work better, we need to control the speed of each motor. The principle is that the Arduino outputs PWM (pulse width modulation) signal, a series of repeating pulses of variable width to the ESCs, and the ESCs make the motors turning at different speeds.

To calibrate an ESC is to set the maximum and minimum speeds of the motor in relation to the max and min width of the PWM signal sent by the Arduino. A PWM signal is simply a square wave signal consisting of high and low (5v and 0v) signals of a certain duration.

The old ESCs, could only spin the motor in one direction under the PWM signal range of 1500us to 2100us. This means while the boats needs to turn, one of the motors will stop and the other one keep pushing the boat.

After calibration, the new ESCs can work on the PWM signal range of 1000-2000 microseconds, and 1500 microseconds is the middle status of the range. Thus, the motor will spin negatively with the PWM signal from 1000 to 1500 microseconds, and positively with the PWM signal from 1500 to 2000 microseconds. With these new ESCs, turning will be faster than before as the two motors rotate in different directions. In that case, the turning radius will be smaller than before. Also with the updates to the software and the help of the cellphone, the boat will know it is stuck and go reverse when it cannot move forward.

At the default setting, the ESC increase and decrease speed is linear (Fig. 5).

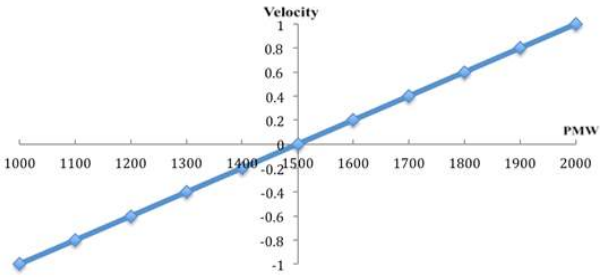


Fig. 5 Default Velocity – PWM Diagram

Unlike the manufacturer's calibration process, we need to calibrate the ESC with an Arduino. The objective of the calibration program is:

1) Keep the input at the maximum pulse-width l (2000 us) with the ESC connected to the battery, until hearing a short series of tones followed by a two second pause, followed by a long series of tones.

2) Keep sending the minimum pulse-width (1000 us) after step 1, until hearing a long series of tones.

3) Keeping sending the middle pulse-width (1500 us) after step 2, until hearing a long series of tones followed by a two second pause, then followed by a short series of tones.

B. Change the Android Program Setting

The Arduino works under the Android Phone's signal. Compared with the old version ESC, the new version one has a totally different reaction in the same situation: the motor will only work with the full input, for example, and the ESCs will activate its own over-current protection system while it is running in the water, to disable the ESC for a few seconds.

To troubleshoot these problems, we changed the safety-thrust number, a factor that restrict the output of the PWM and some related code in the Android program.

After trial and error, we found that 0.14 to be the best safety-thrust for the boat, it makes the boat work sustainably in the water without a too low speed.

C. Program the ESC

Because of these problems, the default setting of the ESC did not meet our demand, and also the Vapor Pro is programmable with the Castle Link Program Kit, which requires a Windows system we needed to change the setting to maximize the efficiency of the ESC.

For example, in default setting, the reverse type is "With Reverse" which requires the controller must be set to neutral throttle for two seconds before it allows reverse operation. So we need to change it to "Crawler Reverse".

Furthermore, the "Throttle Dead Band" setting, which sets the width of the neutral or idle zone, and the default setting is medium. Since the safety-thrust in the Android program is 0.14, so the full speed of the boat is just 0.14 of its full power by restrict the output PWM from the Arduino board. In other words, the changing of the input to the ESC is very small. Thus, we need the width of the neutral or idle zone as small as possible.

Following are the steps of how to program the ESC (Fig. 6):

1) Link an ESC with the PC, and run the Castle Link software. It will load the setting of the ESC automatically.

2) Set it to the default setting if it is a brand new one.

3) Update the firmware of the ESC, if there is an update.

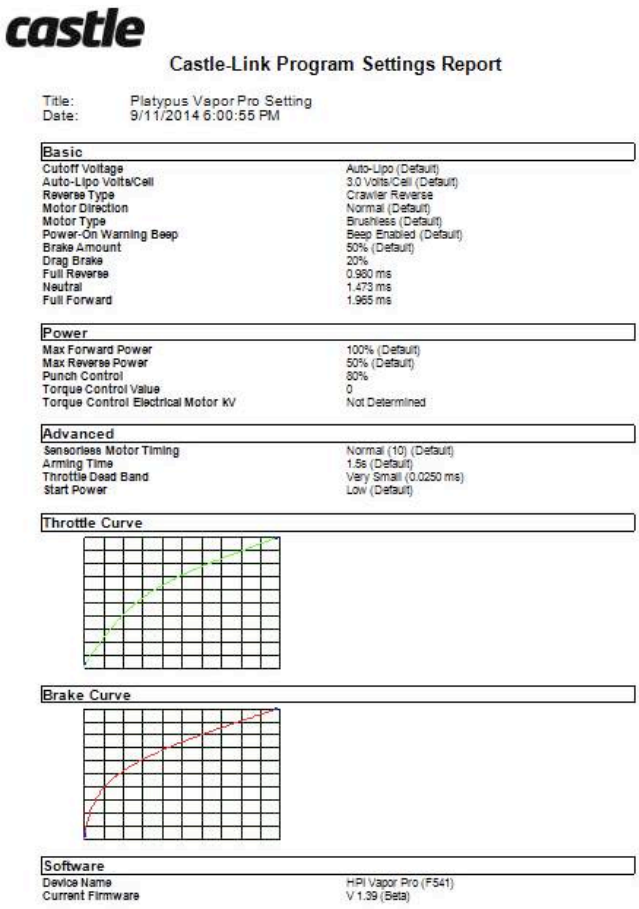
4) Change the Reverse Type to Crawler Reverse.

5) Change the Drag Break to 20%.

6) Change the Punch Control to 80%.

7) Change the Throttle Dead Band to Very Small 0.0250ms.

8) Change the Throttle Curve and Break Curve.


castle

Castle-Link Program Settings Report

Title: Platypus Vapor Pro Setting
Date: 9/11/2014 6:00:55 PM

Basic

Cutoff Voltage	Auto-Lipo (Default)
Auto-Lipo Volts/Cell	3.0 Volts/Cell (Default)
Reverse Type	Crawler Reverse
Motor Direction	Normal (Default)
Motor Type	Brushless (Default)
Power-On Warning Beep	Beep Enabled (Default)
Brake Amount	50% (Default)
Drag Brake	20%
Full Reverse	0.980 ms
Neutral	1.473 ms
Full Forward	1.965 ms

Power

Max Forward Power	100% (Default)
Max Reverse Power	50% (Default)
Punch Control	80%
Torque Control Value	0
Torque Control Electrical Motor kV	Not Determined

Advanced

Sensorless Motor Timing	Normal (10) (Default)
Arming Time	1.5s (Default)
Throttle Dead Band	Very Small (0.0250 ms)
Start Power	Low (Default)

Throttle Curve

Brake Curve

Software

Device Name	HPI Vapor Pro (F541)
Current Firmware	V 1.39 (Beta)

Fig. 6 Vapor Pro Setting Report

IV. FAILSAFE SYSTEM

A failsafe system is a system that runs when the boat cannot connect to the laptop. Mostly it is because the boat is out of range. The Failsafe system will help the boat find it way back home or to some spot where it will get a connection.

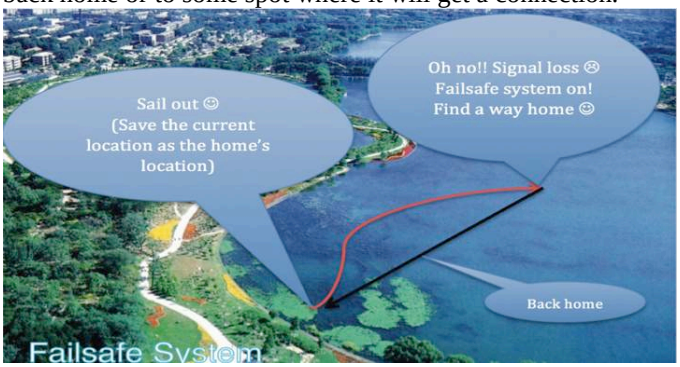


Fig. 7 Failsafe System mechanism

The mechanism of the failsafe system (Fig. 7) and programing thought (Fig. 8) is below. When the failsafe system turns on, the boat will save current location's waypoint before sailing out. After sailing out, the boat turns the GPS sensor on and records the waypoint while the connection is still reachable. Once the connection has been unreachable for 10 seconds and the boat has no path to follow, boats will

immediate go to the last recorded waypoint (which is called the save point) where the boat was able to get the prior connection. After waiting at the save point for 10 seconds (in some circumstances it still is not connected), the boat will head to the original home point directly.

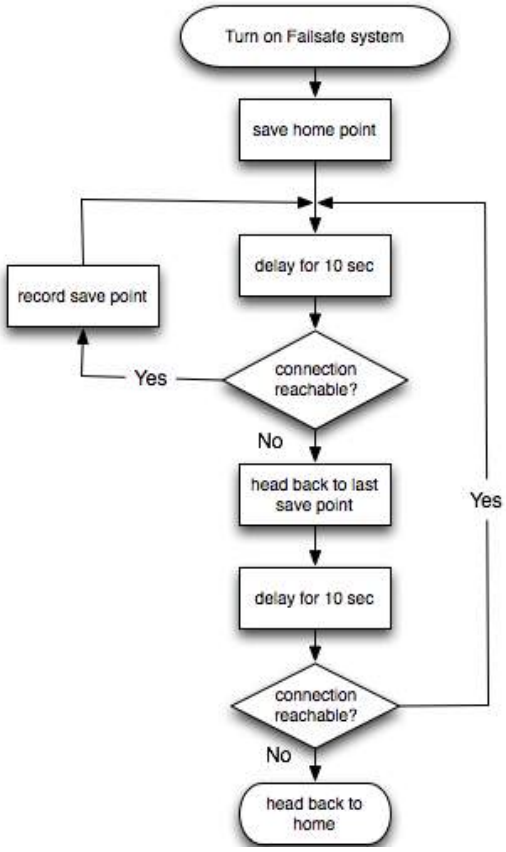


Fig. 8 Failsafe System Flow chart

After several tests, we are sure that this system will make boat more reliable in the water. We do not need to face the risk that we will lose boats because of lost signals.

V. OFFLINE SYSTEM

An offline system (Fig. 9) is an android app that is used to pre-store a sequence of commands in a text file in the cellphone located on the boat. This enables the boat to navigate by itself.

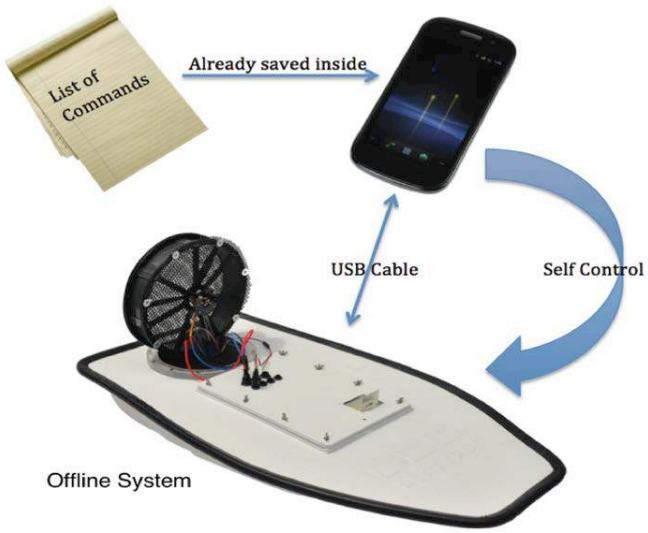


Fig. 9 Offline System Mechanism

In this case, airboats can do self-driving without the limitation of the local network's range.

The boat follows a saved sequence of commands and it can navigate based only on GPS without the help of computers.

One of the main challenges is how to initially turn the system on. The cell phone is inside of boats so it is impossible to add a button to start up. And it is hard to calculate how long it takes to start up the whole airboat firmware thus we cannot add a timer.

The solution is that we use the orientation sensor inside the cell phone and write an algorithm that makes airboats auto-start after manually spin boats around three times.

VI. AUTO DATA FEEDBACK SYSTEM

An auto data feedback system is embedded into the existing Android App. This system helps us avoid the onerous labor of manually transfer the log files to computer by USB.

During each water field test, a log file is created to store data from all the working sensors. It is stored in a specific folder on the phone which path is known. The function of this system is to grab the latest log files base, and when the tester quits the app, it can automatically send an email with the log file attachment to the users' emails. Also a report button is created to allow the tester to report the log file manually.

Data in the log file is what customers care about most, so that is why auto data feedback is critical. The sending email function in this system applies to different email types including Gmail, Hotmail, and so on.

This system can not only give feedback of log files, but also can be used to transfer control commands from the laptop or tablet to the Android phone. So it can later be further applied to the auto interaction between the computer and the Android phone on the boat.

Fig. 10 Auto Data Feedback System

VII. RESULT

A. Electronic Speed Control

After adjusting the ESCs, they are ready to use in real test. We have conducted field testing at Panther Hollow Lake, Pittsburgh, PA. The ESC was able to run in the lake and support all the critical software functions.

B. Failsafe System

The airboat runs smoothly with the failsafe system. The failsafe system increases the reliability of the voyage.

In field testing (Fig.11), once a boat loses its signal, it will come back to the save point in 30 seconds. Disconnections may be caused by the boat running out of range or network errors.

However, because of the poor accuracy of the cell phone's GPS sensor, sometimes the boat may not be able to get to the precise. Accuracy is usually within 5 meters. Thus the boat may stay at the position that is 5 meters off the shoal or just violently hit the river's bank. It should be a focus of future work.

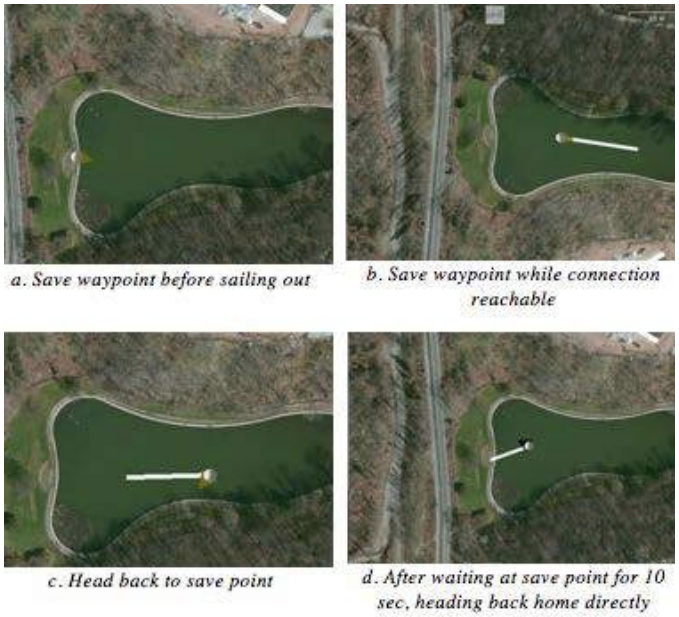


Fig. 11 Failsafe System Result

C. Offline System

The offline system enables the boat to navigate by itself. This worked well in field test. It performed with the same level of accuracy as a laptop to send commands. It is very useful to do some routine tasks like patrolling around a lake for weekly to do some water quality monitoring. In addition, we don't need to carry a heavy tool kit, laptop and network router every time which makes the test much easier than before.

However, we do not yet have an obstacle avoidance algorithm. In some cases the propellers may get stuck in some water weeds or lily pads and stop working, which cause a lot of problems. To make things worse, because it is an offline system, we are not able to control boats manually. Adding some obstacle avoidance algorithm will make it better in the future.

What's more, the boat cannot distinguish whether a command is reachable. For example, the boat might try to get to a destination hundreds of miles away because of a typo or mistake. Thus it will be a great help if it can detect and send warning signal when errors occur.

D. Auto Data Feedback System

Auto data feedback system worked well after field testing. In the tester's Gmail, all the latest log files from the boats after experiments were received.

Gmail and QQ email were used as two different types of customers email types. The initial code ran well to send an email with an attachment to the QQ email but failed in the case of Gmail. Revisions were made on the original so that it can work on Gmail through a stricter security verification. See Fig.10.

Also clicking the report button produces the same result. But in this case, it can report the latest log file not only soon after the test, but also anytime you want after the test.

ACKNOWLEDGMENT

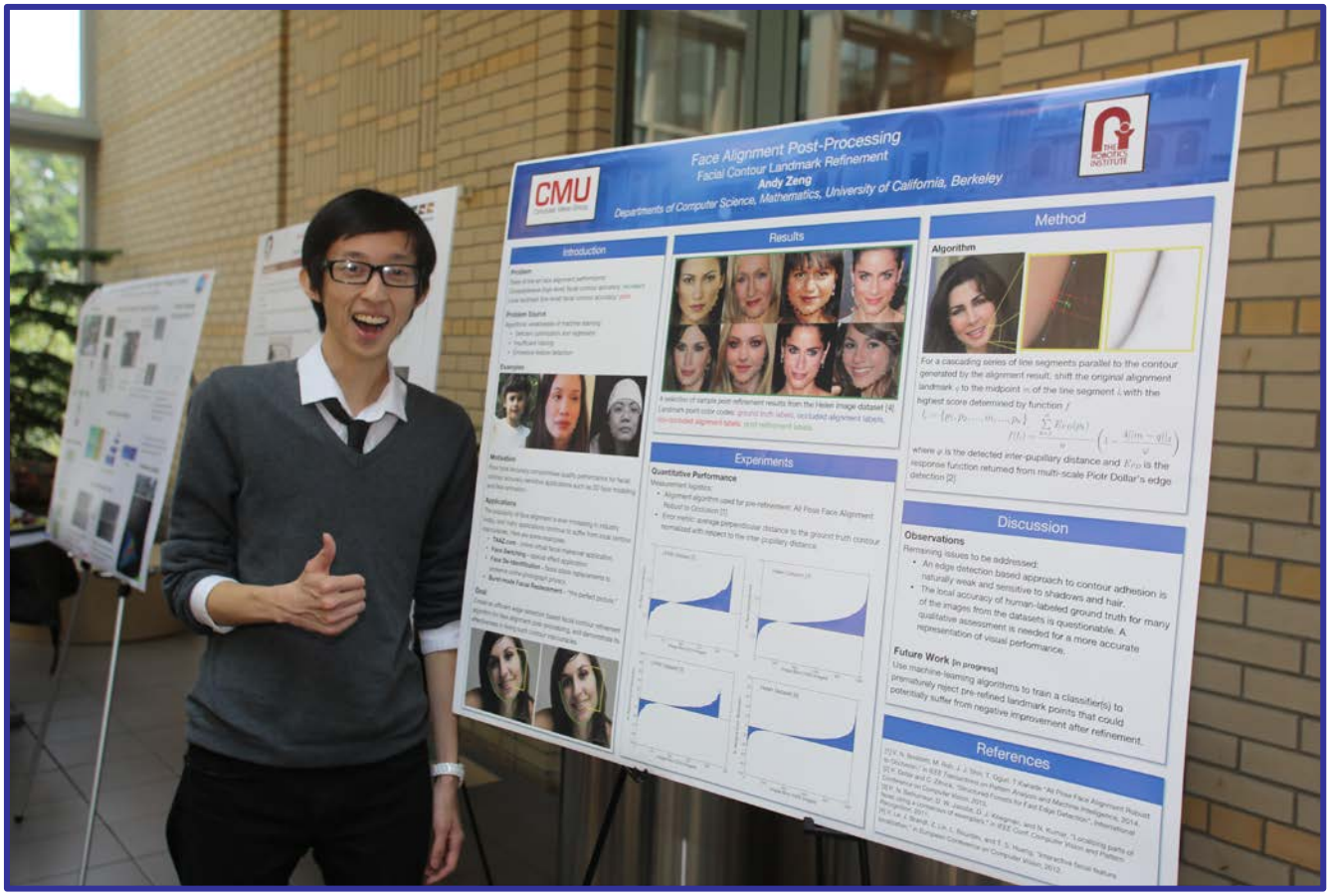
Specially thanks to our supervisor Dor. Paul Scerri for his guidance. Thanks to John Scerri and Nathan Brooks for their valuable suggestions. Thanks to Rachel Brucin and the RISS Program.

REFERENCES

[1] Paul Scerri, Prasanna Velagapudi, Balajee Kannan, Abhinav Valada, Christopher Tomaszewski, John M. Dolan, Adrian Scerri,,Kumar Shaurya Shankar, Luis Lorenzo Bill-Clark, and George A. Kantor, "Real-World Testing of a Multi-Robot Team, "Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012), June, 2012.

Andy Zeng

RISS 2014



Face Alignment Refinement

Andy Zeng Vishnu Naresh Boddeti Kris M. Kitani Takeo Kanade
Robotics Institute, Carnegie Mellon University
andyzeng@berkeley.edu naresh@cmu.edu {kkitani,tk}@cs.cmu.edu

Abstract

Achieving sub-pixel accuracy with face alignment algorithms is a difficult task given the diversity of appearance in real world facial profiles. To capture variations in perspective, occlusion, and illumination with adequate precision, current face alignment approaches rely on detecting facial landmarks and iteratively adjusting deformable models that encode prior knowledge of facial structure. However, these methods involve optimization in latent sub-spaces, where user-specific face shape information is easily lost after dimensionality reduction. Attempting to retain this information to capture this wide range of variation requires a large training distribution, which is difficult to obtain without high computational complexity. Subsequently, many face alignment methods lack the pixel-level accuracy necessary to satisfy the aesthetic requirements of tasks such as face de-identification, face swapping, and face modeling. In many such applications, the primary source of aesthetic inadequacy is a misaligned jawline or facial contour. In this work, we explore the idea of an image-based refinement method to fix the landmark points of a misaligned facial contour. We propose an efficient two stage process - an intuitively constructed edge detection based algorithm to actively adjust facial contour landmark points, and a data-driven validation system to filter out erroneous adjustments. Experimental results show that state-of-the-art face alignment combined with our proposed post-processing method yields improved overall performance over multiple face image datasets.

1. Introduction

Given an estimated facial contour returned from face alignment, our objective is to refine the contour such that it is closer to the true facial boundary. Accurately detecting facial boundaries is a challenging problem because the contours of facial profiles in the real world are subject to a broad range of variation in illumination, occlusion, noise, and individual differences. A facial contour in an image may be partially occluded by hair, faded into the wrinkles, or hidden by shadows. The challenge presented by these problems is further compounded by having to consider the variations in jawline structure and individual facial features that may cause irregularities in the facial outline. Handling such variation at a high level of detail is the key to designing a robust face alignment contour refinement algorithm.

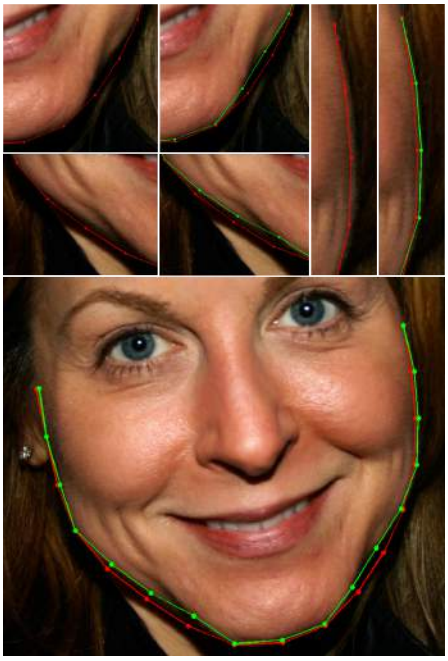


Figure 1: Alignment refinement result. Initial facial contour (red) and refinement result (green).

With the explosive increase in personal photos across the Internet nowadays, the popularity of face alignment in modern applications is rapidly growing. For many of these applications, *e.g.*, face de-identification, face swapping, and face modeling, the aesthetic quality of the aligned facial boundary is quite sensitive to slight misalignment. In the case of face swapping, an estimated facial contour extending past the true facial boundary will introduce background artifacts onto the output face. The main motivation behind the work is an application-side demand for more accurate

face alignment results.

Despite the abundance of research on face alignment, many state-of-the-art methods are not able to align an estimated facial shape to the true facial boundary with sub-pixel accuracy. Face alignment typically involves an optimization problem where the goal is to match some deformable face model to the true face shape as closely as possible using detected facial features as anchor points. During this optimization, the face shape model is often parametrized or constrained, resulting in the loss of the fine detailed information about the facial contour. In this paper, we propose a flexible facial contour refinement method to correct the facial contour inaccuracies of generic face alignment methods using a data-driven post-processing technique.

Based on observations over various state-of-the-art face alignment results [4, 12, 15, 13], we propose a two-step approach for fixing facial contour misalignment. In the first step, we introduce the active adjustment algorithm responsible for shifting individual landmark points that constitute the facial contour. The shifting is performed heuristically based on edge response, the distance from the initial contour estimate returned from alignment, and edge direction. In the second step, we introduce a data-driven validation process that reinforces the overall performance of the active adjustment algorithm by training a classifier to deter the refinement process from making potentially erroneous adjustments.

2. Related Works

Face alignment is a very challenging and well-studied problem. Active Shape Models [6] and Active Appearance Models [5] are the most well known and widely used models for shape-fitting. Constrained Local Models [13, 17, 7] are another class of approaches for face alignment that are largely focused on global spatial models built on top of local landmark detectors. Recently many discriminative shape-based regression approaches [4, 16] have been proposed in the literature. Instead of relying on parametrized appearance and shape models, these approaches leverage large amounts of training data to learn a cascade of regressors, mapping image features to the final facial shape.

The task of refining the contour of a face shape is similar to the problem of contour fitting. Contour fitting generally requires some form of boundary detection, followed by an optimization step, where the fitting of a deformable contour model over the boundaries of interest is performed. Some methods iteratively re-sample adaptive spline models [9, 14] while other methods apply dynamic programming to energy-minimizing deformable contours [1]. The task of facial contour refinement however, differs from the task of contour fitting in that facial contour refinement is given a close initial alignment. Assuming that the results from face alignment return a reasonable estimate of the facial contour, refinement needs to actively work with this information in order to accurately adjust the contour under a wide range of image variation. Furthermore, because the results from boundary detection can sometimes be noisy or misleading, refinement also needs to be conservative in order to minimize the number of erroneous adjustments. In this work, our goal is to construct a refinement algorithm that maximally improves the accuracy of an estimated facial contour only for those images that are problematic.

3. Problem

In a two-dimensional digital face image I, a face shape $S = \{p_i \in \mathbb{R}^2\}_{i=1}^{N}$ consists of N facial landmark points $p_i = (x_i, y_i)$. The goal of face alignment is to estimate a shape S as close as possible to the true shape $\widehat{S}$, *e.g.* to minimize

$$||S - \widehat{S}||_2 \tag{1}$$

Among the N points that constitute a face shape S, there are $M < N$ points that make up the facial contour $C = \{p_i \in S\}_{i=1}^{M}$. Given S, our objective is to fine-tune alignment contour C to be closer to the true contour $\widehat{C} = \{\widehat{p}_i \in \widehat{S}\}_{i=1}^{M}$ after refinement, *e.g.* to maximize

$$\text{Error}(C_{\text{before}}, \widehat{C}, m_p) - \text{Error}(C_{\text{after}}, \widehat{C}, m_p) \tag{2}$$

where m_p is the performance metric, C_{before} and C_{after} are the alignment contours before and after refinement respectively. Equation 2 semantically represents contour improvement, and will be used to guide training and evaluate the performance of our post-processing approach. As part of our objective, we want this value to be as consistently positive as possible.

4. Facial Contour Refinement

In this section, we first introduce our active observation-based adjustment process. Conceptually, the algorithm individually adjusts each landmark point p_i of a given alignment facial contour $C \subset S$ by shifting it to the nearest, strongest edge that is closely parallel to the facial outline originally generated by alignment contour C. But since this method is constructed on the basis of human intuition, it remains incapable of performing robustly under the wide range of misalignment variations in illumination, noise, occlusions, etc. Hence, if used without proper discretion, this algorithm is susceptible to performing "bad" adjustments. Therefore, we present a compatible data-driven validation framework, in which we conditionally perform the active adjustments based on prior post-refinement observations. Given that each facial contour consists of M landmark points, we train M distinct SVMs over a large collection of training faces in order to be able to determine, per contour at test time, which alignment landmark points

should undergo active adjustments and which points should be left alone in order to maximize the overall contour improvement. As we shall see later, this form of preemptive filtering is necessary in our refinement approach in order to conceivably maintain positive contour improvement as consistently as possible.

4.1. Active Refinement

Our active adjustment algorithm is based on three major observations. The first observation is that a true facial boundary is more likely to be located on an edge than anywhere else. Given a landmark point p_i in facial contour C, a naïve adjustment algorithm reflecting this observation would shift p_i to the point with the strongest edge response within some small search radius r, *e.g.*

$$P = \{s \in \mathbb{R}^2 : ||p_i - s||_2 < r\}$$
$$f_h(p) = \text{Edge}(p)$$
$$\text{Refine}(p_i) = \{p \in P \text{ s.t. } f_h(p) = \max_{s \in P} f_h(s)\} \quad (3)$$

where $\text{Edge}(x) \in [0, 1]$ returns the edge response for a point x. Our second observation is that the pre-refined alignment facial contour serves as an adequate estimate for the true facial boundary. To incorporate this into the first observation reflected in Eq. 3, we add a distance factor to the heuristic function $f_h(p)$, *e.g.*

$$f_h(p) = w_1 \text{Edge}(p) + w_2 \left(1 - \frac{||p_i - p||_2}{r}\right)$$
$$\text{Refine}(p_i) = \{p \in P \text{ s.t. } f_h(p) = \max_{s \in P} f_h(s)\} \quad (4)$$

where w_1 and w_2 are weights. The refinement algorithm in Eq. 4 using the new heuristic function, as it currently stands, may adjust landmark points to edges that do not retain the innate facial structure estimated from alignment. In other words, the variation in edge direction is not properly constrained, *i.e.* erroneously shifting a landmark point around the chin to the edge of a collar directionally perpendicular to the outline generated by the true facial contour. So our final observation, conceptually derived from our second observation, is that the outline generated by pre-refined alignment facial contour should be near parallel to the outline generated by the true facial boundary.

Under all three observations, the ultimate goal of our active refinement algorithm is to move each landmark point to the nearest, strongest edge segment that is near parallel to the outline generated by the alignment facial contour. See Figure 2. More specifically, for each alignment landmark point p_i in facial contour C, we generate a series of cascading line segments parallel to the outline generated by C, where each line segment is explicitly defined as a collection of points in a single direction *e.g.*

$$v_{i_c} = p_{i+1} - p_{i-1} \qquad v_{i_p} = (-v_{i_{c_y}}, v_{i_{c_x}})$$

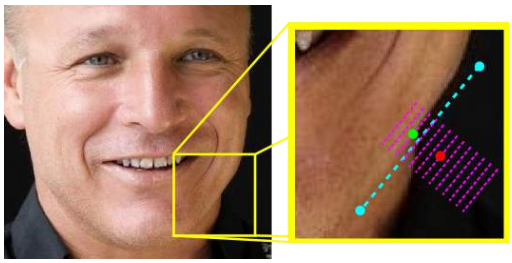


Figure 2: Active adjustment. The landmark point (**red**) is updated to a new location (**green**) by searching along a line perpendicular to the tangent line (dashed cyan line).

$$b_i = p_i - \frac{r}{2}\left(\frac{v_{i_p}}{||v_{i_p}||}\right) - \frac{r}{2}\left(\frac{v_{i_c}}{||v_{i_c}||}\right)$$
$$p_{k_j} = b_i + \frac{r}{\sigma}k\left(\frac{v_{i_p}}{||v_{i_p}||}\right) + \frac{r}{\sigma}j\left(\frac{v_{i_c}}{||v_{i_c}||}\right)$$
$$L = \{L_k\}_{k=1}^{\sigma} \qquad L_k = \{p_{k_j}\}_{j=1}^{\sigma} \quad (5)$$

where p_{i+1} and p_{i-1} are neighboring alignment contour points of p_i, and σ is a saturation value. A score is computed for each line segment in L based on the heuristic function $f_h(p)$ in Eq. 4

$$\text{score}(L_k) = \sum_{j=1}^{\sigma} \frac{f_h(p_{k_j})}{\sigma} \quad (6)$$

and p_i is shifted to the midpoint of the line segment with the highest computed score from Eq. 6, *e.g.*

$$L_{\text{best}} = \{L_k \in L \text{ s.t. } \text{score}(L_k) = \max_{l \in L} \text{score}(l)\}$$
$$L_{\text{best}} = \{p_{\text{best}_j}\}_{j=1}^{\sigma}$$
$$\text{Refine}(p_i) = p_{\text{best}_{\lceil \frac{\sigma}{2} \rceil}} \quad (7)$$

4.2. Data-Driven Validation

An observation-based edge detection approach to refinement is sufficient to fix the easy misalignment cases. However, a large percentage of misaligned landmark points is still difficult to assess and fix, even to the human eye. We address this problem by adopting a data-driven approach to recognize and preemptively avoid such difficult cases. For each of the M landmark points that make up a facial contour C, we train a binary SVM to classify each corresponding landmark as an easy or difficult case. In order to minimize the total number of erroneous adjustments, these classifiers are used to limit the refinement algorithm from adjusting the difficult cases.

Contrast normalized pixel values extracted from a small region around each point serve as the features used to train the classifiers. These patches are rotated with respect to the outline generated by the alignment facial contour such that

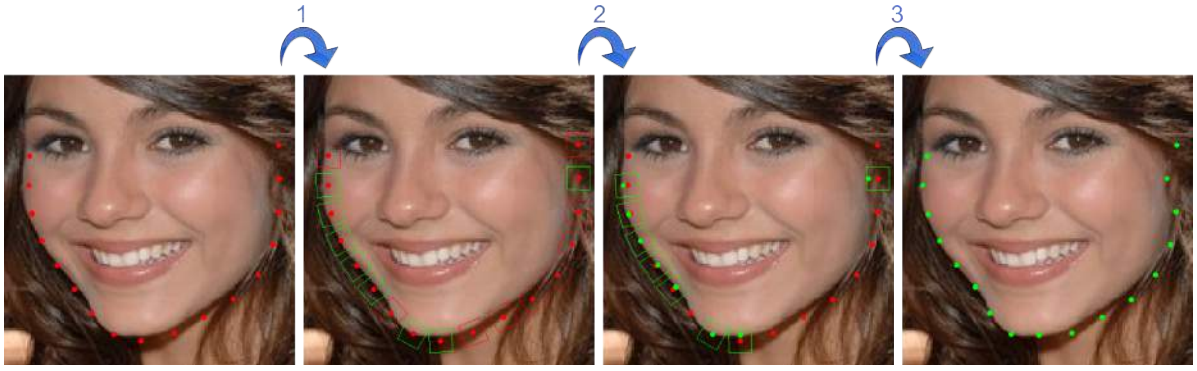


Figure 3: Refinement pipeline: (1) select landmark patches to be updated (green boxes), (2) adjust landmark positions, (3) return updated landmarks.

the right side of the patch is further away from the face than the left side. At test time, prior to running the active adjustment algorithm for each landmark point, we generate a feature vector reflecting the small rotated square region around the point. If the SVM classifies the region as likely to facilitate negative improvement after refinement, then we skip active adjustment for that particular landmark point. This is done before the adjustment of each landmark point of every test facial contour. Since our objective for face alignment post-processing is to maintain positive improvement as consistently as possible, this validation framework essentially reinforces the dependability of the active adjustment algorithm by learning and avoiding potential erroneous adjustments. Figure 3 illustrates the role of the validation framework within the refinement approach.

5. Experiments

In this section, we provide some experimental analysis which highlights the advantages of our proposed facial contour refinement approach. The experiments are designed to demonstrate the validity of our active adjustment method, illustrate the intuition behind the validation framework, and evaluate the quantitative and qualitative performance of our refinement approach as a whole.

Face Alignment For our experiments, we use a face alignment method, based upon [13, 3], that is robust to occlusions and approximates face shape S and returns $\mathbb{N}$ binary labels corresponding to the estimated state of occlusion for each individual landmark point in S. Since our approach was not designed to be robust for occluded landmark points, during refinement we limit our adjustments to the non-occluded points in order to minimize the number of misalignment cases attributed to occlusions.

Datasets We demonstrate the efficacy of our contour refinement approach, by evaluating it on three different face datasets namely, HELEN [11], "Labeled Face Parts in the Wild" (LFPW) [2] and "Annotated Faces in the Wild" (AFW) [17]. For consistent cross-database annotations, we used the generated annotations provided by IBUG [10]. Each dataset presents a different challenge due to varying degrees of image quality and facial variation. The HELEN dataset contains 2,000 training and 330 testing high resolution face images obtained from Flickr. The LFPW face image dataset features 811 training and 224 testing images pulled from the Internet using simple search queries. The facial images in both of these datasets exhibit a wide range of appearance variations including pose, lighting, facial expressions, occlusion, and individual differences. The AFW dataset consists of 337 face images out of which we isolate 100 images for testing, and use the rest for training. Most of the faces from this dataset have poor image quality and/or low resolution and consist of faces captured under unconstrained conditions. For all datasets, images where the primary face is undetected using [13] are excluded from our experiments.

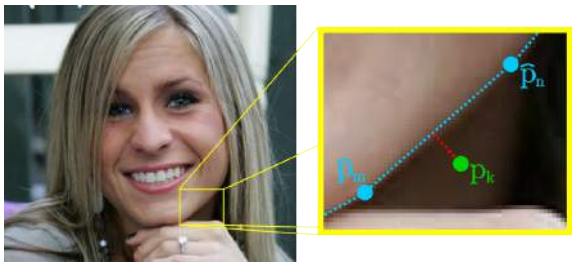


Figure 4: A conceptual visualization of the error metric introduced in Equation 9. The accuracy of some point $p_k \in C$ is measured by its distance to the facial outline generated by the true contour $\widehat{C}$ (blue).

Edge Detection In our experiments, we use the fast edge detection method proposed by Dollár and Zitnick [8]. Capable of multi-scale edge detection, this edge detector features superior run-time complexity while maintaining state-

of-the-art edge detection performance.

Error Metric If we derive from the metric in Equation 1, then the facial contour error is computed as

$$\text{Error}(C, \widehat{C}) = ||C - \widehat{C}||_2 \tag{8}$$

The relative position of each estimated landmark point p_i on the alignment facial contour C is likely to differ from the relative position of its ground truth counterpart $\widehat{p}_i$ in $\widehat{C}$. Unfortunately, the error metric in Eq. 8 does not capture this variation. So we adopt a new error metric m_p that conceptually reflects the perpendicular distance between each landmark point of C to the outline generated by the true contour $\widehat{C}$ (see Fig. 4), *e.g.*

$$\text{Error}(p_k, m_p) = \frac{|(\hat{p}_n - \hat{p}_m) \times (\hat{p}_k - \hat{p}_m)|}{\widehat{\varphi}||\hat{p}_n - \hat{p}_m||_2} \tag{9}$$

$$\text{Error}(C, \widehat{C}, m_p) = \sum_{i=1}^{M} \text{Error}(p_i, m_p) \tag{10}$$

where $\hat{p}_n, \hat{p}_m \in \mathbb{R}^2$ denote the two landmark points in $\widehat{C}$ closest to p_k via Euclidean distance and $\widehat{\varphi}$ represents the true inter-pupillary distance. We use the per-point error metric in Equation 9 to guide validation training, and the relative accuracy improvement metric (semantically defined as the % error reduced after refinement)

$$\frac{\text{Error}(C_{\text{before}}, \widehat{C}, m_p) - \text{Error}(C_{\text{after}}, \widehat{C}, m_p)}{\text{Error}(C_{\text{before}}, \widehat{C}, m_p)} \tag{11}$$

and the absolute accuracy improvement metric

$$\text{Error}(C_{\text{before}}, \widehat{C}, m_p) - \text{Error}(C_{\text{after}}, \widehat{C}, m_p) \tag{12}$$

in conjunction with Eq. 11 to evaluate the performance of our post-processing approach.

5.1. Performance of observation-based refinement

In section 4.1, we described the three major observations around which our active adjustment algorithm is structured. Recall that with each major observation, we intuitively modified our adjustment algorithm to reflect that observation. In this experiment, we demonstrate how the integration of each modification works to boost the overall performance of our refinement approach. We train and test the complete refinement approach three times while swapping out the adjustment algorithm each time; once using Eq. 3 (one incorporated observation), once using Eq. 4 (two incorporated observations), and once using Eq. 7 (all three incorporated observations). Since the training of the data-driven validation framework learns from the prefiltered performance of the active adjustment algorithm over the training image dataset, the adjustments algorithms are changed before each training session to reflect the swapped adjustment equations used respectively during testing. Additionally, it is important to note that even though we used the Helen dataset to generate the results of this experiment in Fig. 5, the results generated by the LFPW and AFW datasets were consistently similar. For this experiment, as well as the following experiments, we empirically chose the active adjustment parameters to be $\sigma = 5$, $r = \varphi/4$ (where φ is the detected inter-pupillary distance) for a good trade-off between accuracy and computational cost.

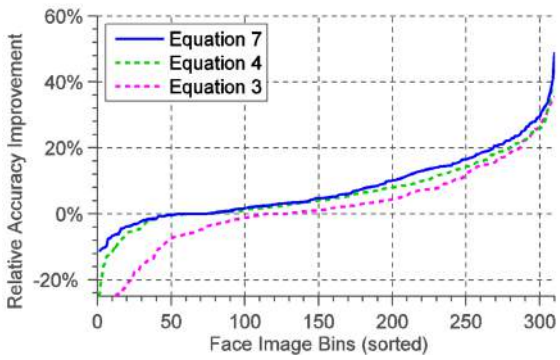


Figure 5: Comparison of different active adjustment algorithms in Equation 3, Equation 4, and Equation 7. Relative accuracy improvement is based on Equation 11.

Figure 5 illustrates the relative accuracy improvement (Eq. 11 with Eq. 10) of every test face (sorted by improvement). We see that incorporating the second observation made in section 4.1 to formulate Eq. 4 worked very well in reducing the number of cases where the naive implementation using Eq. 3 would have inaccurately shifted a landmark point to an outlier edge. Reducing the search space effectively reduced the possibility of misalignment. Additionally, we see that the active adjustment algorithm using Eq. 7 was able to further reduce some of the outlier misalignment cases attributed to the edge direction variation. Note that incorporating all three observations made in section 4.1 yields the best overall performance.

5.2. Verifying the data-driven validation framework

For this experiment, we verify the effectiveness of the data-driven validation framework used to reinforce the overall performance of the active adjustment algorithm.

Parameter Settings In our experiments, the C-SVC SVM was trained with the polynomial kernel $K(u, v) = (\gamma u^T v + c)^d$ where the parameters were empirically chosen as $\gamma = 2$, $c = 1$, $d = 3$ for consistent cross-dataset performance. For both training and testing, given detected face shape width w, the size of the localized square patches around each landmark point were set at $l \times l$ pixels where $l = \frac{w}{10}$.

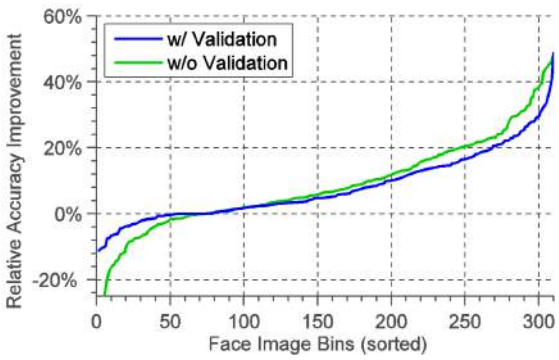


Figure 6: Comparison of refinement with and without the data-driven validation framework described in section 4.2. Relative accuracy improvement is based on Eq. 11.

Figure 6 presents a comparison between the refinement results with and without the data-driven validation framework. We see that although the accuracy improvement of some of the face shapes were not as high with the validation process as they were without it, nonetheless the total number of erroneous adjustments was significantly reduced with the validation process. The results of the refinement with the validation framework are much more desirable since our objective is to improve alignment accuracy through active adjustment while reducing the total number of erroneous adjustments as much as possible.

Visualizing the validation framework

Figure 7 illustrates the localized patches around landmark points with the lowest and highest decision values from the SVM, and their weighted averages. From the visualization of the averages, we see that the SVM learns to avoid adjusting landmark points that are already located near a strong gradient (presumed to be the true facial contour). On the other hand, the SVM also learns to favor adjustments to be made for points that are only slightly off from a strong gradient, since the active adjustment algorithm is more likely to successfully improve the accuracy of a smaller case of misalignment.

It is also interesting to note that since the patches are localized such that the right side of the patch is further away from the face than the left side, the average patches over the highest decision values seem to imply that better adjustments are made for points that located away from the face, as opposed to points that lie directly on the face. This makes sense, because an edge detection based adjustment algorithm is much more likely to fail due to wrinkles, facial hair, or other similar edge-like facial features. This is why the validation framework is important and necessary to minimize the possibility of erroneous adjustments.

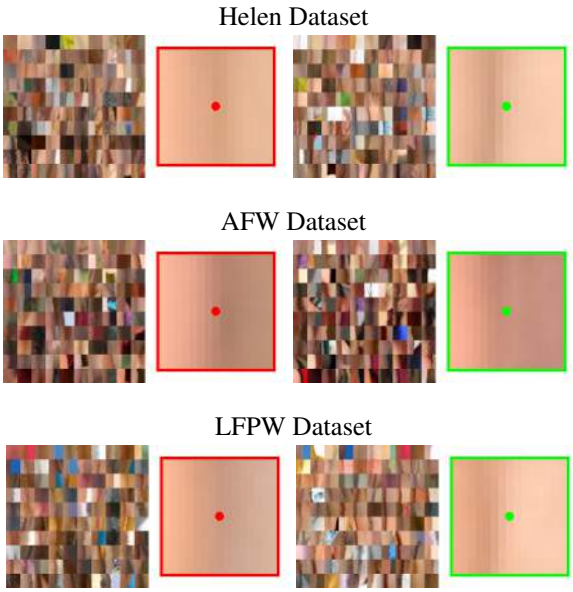


Figure 7: For each dataset, the left mosaic presents the top 100 patches around the landmark points with the lowest decision values from the SVM. The patch highlighted in red is the weighted average over the pixel intensities of all patches in the left mosaic. Similarly, the right mosaic presents the top 100 patches around the landmark points with the highest decision values, and the patch highlighted in green is their weighted average over pixel intensities.

5.3. Quantitative Evaluation

In this experiment, we directly evaluate the absolute contour accuracy improvement of the refinement method over all three datasets. Fig. 8 illustrates the average absolute accuracy improvement for each individual landmark point for every face in each dataset after refinement. Landmark points skipped by the validation framework (and hence have no accuracy improvement) are excluded from these bar graphs. Note that refinement does reasonably well in the Helen dataset where image quality and ground truth annotation accuracy are both high. Refinement performs quite consistently with the LFPW dataset. And finally, as expected, refinement did not do so well in the AFW dataset, where ground truth annotations lacked sub-pixel accuracy, and image quality was sometimes very low (featuring some faces with widths $<$ 200 pixels). Relative differences in accuracy improvement between landmark point indices can reflect the structural weaknesses of the deformable facial models being optimized during face alignment. Overall, the refinement process generally does well to improve the accuracy of face alignment [13]. Table 1 summarizes the computed average contour error (Eq. 12) over every face for each dataset before and after refinement.

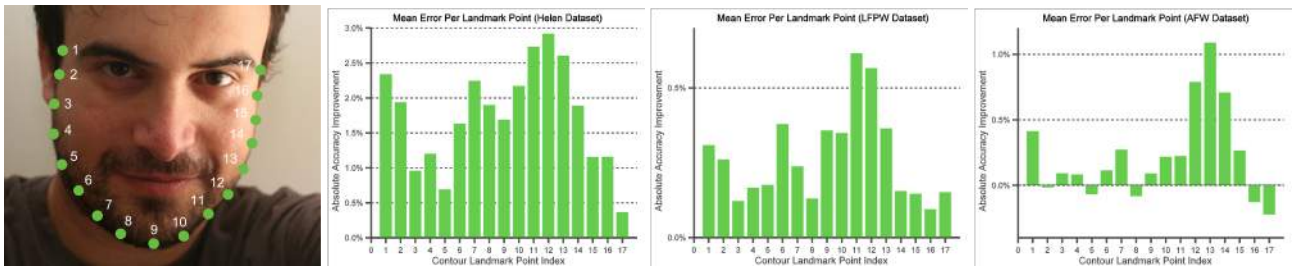


Figure 8: Comprehensive quantative results over the Helen[11], LFPW [2], and AFW [17] datasets. One the left we show the 17 landmark points from alignment [13]) used to describe the facial contour and the graphs display the average absolute accuracy improvement (Equation 9) for each landmark point over all test images in each dataset.

Table 1: Mean Contour Error of [13] (in %)

Dataset	Before Refinement	After Refinement
Helen [11]	3.6639	**3.4587**
LFPW [2]	3.4083	**3.2720**
AFW [17]	4.9339	**4.8610**

5.4. Qualitative Evaluation

It is important to keep in mind though that the quantitative experiments may not be completely representative of the true performance of the refinement approach. The main motivation behind the construction of this method was, after all, to improve upon the aesthetic quality of state-of-the-art face alignment results. Furthermore, the ground truth annotations provided by the datasets that we used were built for the purpose of evaluating face alignment performance, where sub-pixel accuracy for each and every landmark point is typically not to be expected, especially for very high resolution images. Therefore, our experiments require a qualitative evaluation to give a better picture of overall aesthetic improvement in face alignment results after refinement. For each test face image across all datasets, we generate a copy of the face image where the face alignment contour points before refinement are highlighted in red and the shifted contour points after refinement are highlighted in green. See Figure 9. Table 2 summarizes the average results of our questionnaire, where 3 subjects are asked to step through all test face images of each dataset, and judge whether or not the contour improved after refinement. If no contour change was observed, or if there was some difficulty in discerning the state of contour improvement, the subjects were asked to mark 'uncertain' on the questionnaire.

We see that for most test face images from the Helen and LFPW datasets, the subjects noticed an improvement in the accuracy of the facial contour. However, for the AFW dataset, the subjects had some difficulty in judging whether or not there was improvement - this is likely due to the fact that this dataset contains many images with faces that have a low resolution. Overall, the qualitative tests overwhelmingly suggest that our refinement approach facilitates an improvement in the aesthetic quality of face alignment results.

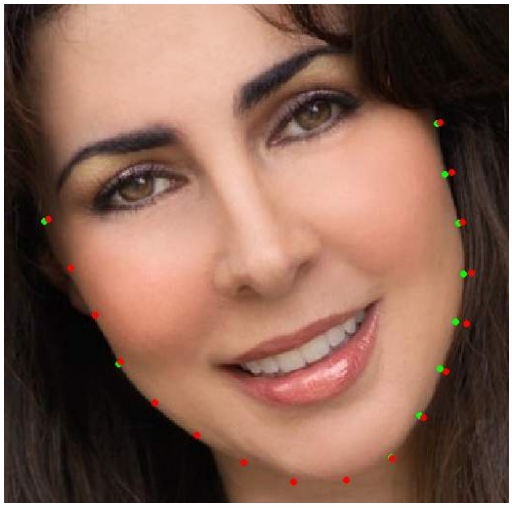


Figure 9: Sample refinement visualization (image from LFPW with the original alignment (red) and refined alignment (green)) shown to subjects during qualitative evaluation.

Table 2: Contour Improvement: Qualitative Evaluation (%)

Dataset	Yes	No	Uncertain
Helen [11]	90.2516	5.3459	4.4025
AFW [17]	59.6667	6.3333	34.0000
LFPW [2]	95.8333	2.7778	1.3889

Finally, to gauge some of the potential improvements that our post-processing method can bring to certain face alignment applications, we constructed a naïve automatic face swapping algorithm and compared the results using the original face alignment points to the results using the refined face alignment points. A sample result is provided

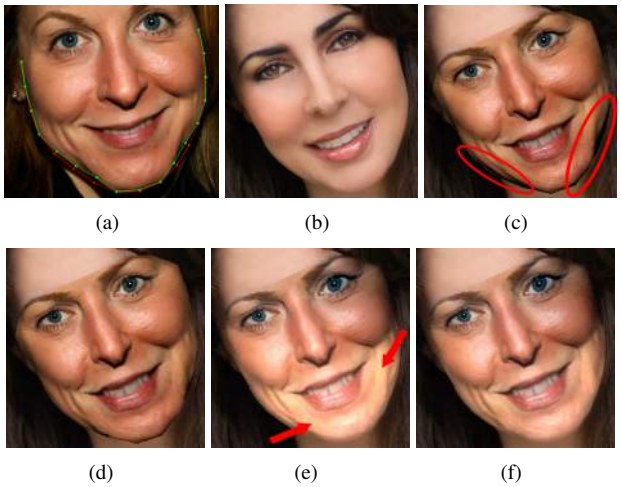


Figure 10: A face swapping example where the sample face and its contour refinement result from Figure 1 (a) is warped into background face (b) using face alignment point without refinement (c) and with refinement (d), followed by Poisson blending over the face replacement results using face alignment without refinement (e) and with refinement (f).

in Figure 10. Note the differences in facial illumination between image (e) and image (f). The dark regions of the background captured by the misaligned facial contour force the bottom half of the warped face to be discolored after Poisson blending. Using refinement to reduce facial contour misalignment effectively reduces the facial boundary noise that can affect face swapping results.

6. Discussion

We proposed an observation-based active adjustment algorithm to fix the inaccurate landmark points of a given contour from a face shape returned from face alignment. To reinforce the performance of this algorithm, we introduced a data-driven validation framework to learn the weaknesses of the algorithm and to minimize the number of erroneous adjustments from refinement. Our evaluation demonstrates that our approach is capable of consistently improving the sub-pixel accuracy as well as the aesthetic quality of a given facial contour. The active adjustment algorithm can also be applied to other problems like object contour refinement and structure segmentation boundary refinement.

References

[1] A. A. Amini, T. E. Weymouth, and R. C. Jain. Using dynamic programming for solving variational problems in vision, 1990. IEEE Transactions on Pattern Analysis and Machine Intelligence.

[2] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars, 2011. IEEE Conference on Computer Vision and Pattern Recognition.

[3] V. N. Boddeti, T. Kanade, and B. Kumar. Correlation filters for object alignment. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[4] X. Cao, Y. Wei, F. Wen, and J. Sun. Face alignment by explicit shape regression, 2012. IEEE Conference on Computer Vision and Pattern Recognition.

[5] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[6] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[7] D. Cristinacce and T. Cootes. Automatic feature localisation with constrained local models. *Pattern Recognition*, 41(10):3054–3067, 2008.

[8] P. Dollár and C. L. Zitnick. Structured forests for fast edge detection, 2011. IEEE International Conference on Computer Vision.

[9] M. A. T. Figueiredo, J. M. N. Leitao, and A. K. Jain. Adaptive b-splines and boundary estimation, 1997. IEEE Conference on Computer Vision and Pattern Recognition.

[10] IBUG. http://ibug.doc.ic.ac.uk/resources/300-W/. http://ibug.doc.ic.ac.uk/resources/300-W/.

[11] V. Le, J. Brandt, Z. Lin, L. Boudev, and T. S. Huang. Interactive facial feature localization, 2011. European Conference on Computer Vision.

[12] L. Liang, R. Xiao, F. Wen, and J. Sun. Face alignment via component-based discriminative search, 2008. European Conference on Computer Vision.

[13] M.-C. Roh, T. Oguri, and T. Kanade. Face alignment robust to occlusion. In *Automatic Face & Gesture Recognition and Workshops*, 2011.

[14] D. Rueckert and P. Burger. Contour fitting using an adaptive spline model, 1995. British Machine Vision Conference.

[15] G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. Robust and efficient parametric face alignment, 2011. IEEE International Conference on Computer Vision.

[16] X. Xiong and F. De la Torre. Supervised descent method and its applications to face alignment. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[17] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

Carnegie Mellon
THE ROBOTICS INSTITUTE

www.ri.cmu.edu/RISS