Moving The Sailing Stone

Enas Tarawneh*, David Perrett†, Jaspal Singh*, Robert Codd-Downey*, Masoud Hoveidar Sefid*, Shreyansh Jain Jeetmal* and Michael Jenkin*
*Lassonde School of Engineering, York University, Toronto, Canada
† AMPD, York University, Toronto, Canada

Email: (enas, jaspal, robert, hoveidar, jenkin)@eecs.yorku.ca, jshreyanshjain@gmail.com

Abstract—The Sailing Stone project animates a stone sculpture such that a traditional, large scale, normally static installation physically interacts with the public. The Sailing Stone moves so as to disrupt the normal motion of visitors to the space so as to encourage changes in the nature of the interaction between visitors and the sculpture itself. The public gallery space is monitored through a network of video cameras, and this information is used to develop a model of human motion through the gallery space. Based on this model of visitor motion, a motion plan is developed and executed for the sculpture so as to maximally disrupt the motion paths of visitors.

Keywords—robot as art, human-robot interaction.

I. INTRODUCTION

Although traditional art is static in that it does not physically respond to the viewer a number of more contemporary artists are exploring interactive art which deviates from this traditional model (see [1] for example). The Sailing Stone project is a robotic work of art which fuses sensory tracking of people in a public space with autonomous exploratory movement, and a stone sculpture. In part, this project explores how viewers interact with traditionally static installations by having the installation move in response to the motion of viewers throughout the exhibition space. In this particular installation, the sculpture is designed to move very slowly – sufficiently slowly so that its motion does not capture the eye of the viewer – in such a way as to disrupt the normal behaviour of viewers. Developing such an installation requires solutions to a wide range of problems related to the art itself and also solutions to a range of technical problems associated with monitoring the gallery space to identify where best to move the robot to disrupt the normal traffic flow as well as moving the robot in a controlled manner. This paper describes ongoing efforts to address these technical problems. Specifically, this paper describes solutions for the Sailing Stone to monitor the gallery space so as to determine flow patterns through the space and then planning and executing motions of the statue so as to change the way in which visitors interact with the gallery space.

The intended movement of the robot is to be glacially slow so that the movement of the robot to casual observer is only apparent after repeated visits to the space. The sculpture is intended to operate as a subversive, agentic presence in human space so that the robot’s movements encourage viewers to re-examine how they interact with spaces. By specifically making the robot’s chosen locations inconvenient, the goal is to encourage questioning of the objects and places that are taken for granted and relegated to the blur of sensory input that is normally blocked out. The ultimate goal of the Sailing Stones project is to exhibit it across Canada both in Artist Run Centres and in public venues. As a consequence, sensing and motion systems need to be adaptable to new locations and obstacles with relative ease, and be able to scale up or down to suit varied presentation venues.

Physically, the Sailing Stone sculpture is as a large, hollow stone block roughly 1 m high, 1.5 m long and massing roughly 800 kg. The basic structure of this stone is shown in Figure 1. This sculpture is mounted on a traditional differential drive robot platform (Figure 1(a)). The base itself is invisible to gallery viewers and enables the normally stationary statue to move about the gallery. The robot base within the Sailing Stone is augmented with a laptop computer to provide local communication with the robot base and to provide a WIFI link to offboard sensing, computation and control. The structure of the Sailing Stone precludes onboard sensors to aid in navigation and pose estimation tasks, so all such tasks are performed using off-board sensors as described later in this paper.

Safe motion of the Sailing Stone requires the robot to be able to sense the free space of the gallery and the presence of people within it. In order to disrupt as little as possible the nature of the sculpture itself, sensors are not placed on the robot itself. Rather, the gallery is augmented with a network of video cameras that obtain views of the gallery space. These cameras capture data about the cumulative travels of people in the sculpture/robot’s surroundings.

Data from the camera network is processed in order to identify visitors’ “desired paths” through the space. That is to identify the common lanes of travel that the majority of people follow as they visit the gallery. After analysis of the pedestrians in the space, the robot then moves very slowly to a location in the exhibit space that maximally disrupts...
the flow of the visitors while not re-visiting places that the
robot has visited previously. The goal here is to disrupt the
normal manner in which visitors interact with the gallery
space and the art within it.

The sailing stone project is a “work in progress” and
this paper describes the state of the project as of early
summer, 2016. Further experiments with the Sailing Stone
are planned with the eventual goal of displaying the Sailing
Stone in a public exhibition later in the year.

II. PREVIOUS WORK

There is a long history of the interaction of autonomous
systems and art. Indeed, many of the very earliest au-
tonomous systems were primarily intended to meet artis-
tic demands, rather than meeting functional requirements.
Mechanical devices designed primarily for their artistic or
entertainment value can be found in many cultures. Early
examples include the Japanese karakuri ningyo (mechanical
dolls) from the Edo Period (1603–1868)[2] and the animated
dolls of Pierre Jaquet-Droz from the 18th century[3]. The
Book of Knowledge of Ingenious Mechanical Devices by Ibn
al-Razzaz al-Jazari in 869[4] describes a range of mechanical
devices whose primary purpose is to entertain and delight.

With the advent of digital electronics more sophisticated
electromechanical devices became possible. Digital robots
have been involved in a range of artistic endeavors from
robot sculpture (e.g., [5]) to dancing quadcopters (e.g., [6]).
There have also been a number of efforts to develop robotic
statues that move about their gallery space. For example, [7]
describes the Diamandini Interactive humanoid robot project
that was exhibited at the Victoria and Albert Museum. This
robot was humanoid in shape and moved about the gallery
space in order to have individuals interact with it. The goal
of the robotic art described here is somewhat different. The
Sailing Stone is not intended to resemble a robot in any way.
Rather, the goal is for the Sailing Stone to appear to be inert
and to move surreptitiously at speeds that are sufficiently low
that observers are unaware of the motion.

III. OVERVIEW

Most autonomous robot systems base the bulk of the
computation and sensing hardware on the robot itself, in
order for the device to be as independent as possible from the
surrounding environment. The Sailing Stone project requires
that the robot be obscured from the gallery by the stone
itself, and thus the robot in the stone must rely on sensors
mounted about the gallery, rather than on the robot. The
primary software requirement of the project is to enable the
construction of “activity maps” of the gallery. That is, to
construct maps of the trajectories of visitors to the gallery.
This requirement leads to the need to be able to image
all of the interesting open space in the gallery. Given the
potential size of a gallery space and its likely layout, it is
unrealistic to assume that a single video camera will be able
to capture all of the interesting ground space. This requires
that the Sailing Stone image capture system integrate data
over multiple cameras and transform this data into a single
consistent representation. Note that it is not practical to
merge the raw video feed of all of the cameras into a single
“integrated image” as the size of any such image could easily
overwhelm the computational resources associated with the
project. In order to overcome this, it was decided to process
individual camera images separately and then to integrate
the results of these processes at a higher level.

In terms of sensing capabilities, localization of the robot is
also a critical issue. Given that the robot is hidden inside the

Figure 1. The Sailing Stone. (a) Shows the Stone being lowered onto its robot base. (b) shows the Stone resting on the base. When the Sailing Stone is
mounted on the base as shown in (b) the robotic base is completely hidden from view. The robot base is controlled by an on-board laptop which connects
to external computation and a sensor network via WIFI.
stone localization must also rely on off-board sensors. The camera network used to identify activity within the gallery space is also used to localize the robot within the space and correct for odometry errors.

In order to simplify the process of analyzing individual images and integrating their results, the image processing infrastructure utilizes standard computer vision software (OpenCV [8]) implemented within a Robot Operating System [9] framework. The choice of ROS and OpenCV provides a large range of standard libraries and toolkits to simplify the process of moving the A constellation of standard web cameras mounted around the exhibition space is used to localize both the robot within the space as well as to identify motion paths through the environment.

Finally, it is necessary to move the Sailing Stone about the space. For the Sailing Stone we re-purposed a TRC LabMate[10] differential drive motion base. Safe motion of an autonomous agent in a crowded space is a difficult task. Fortunately here we are planning on moving the robot at extremely slow speeds thus reducing the need for sophisticated detection or response to dynamic agents (people) in the environment.

IV. MOVING THE SAILING STONE

Actually moving the Sailing Stone involves building two representations of space, the first is a description of the motion of visitors in the free space of the gallery over a given time window. Determining this “activity map” involves integrating motion data from visitors to the gallery from the camera network. The second map represents where the robot has been. Intermittently, these two maps are evaluated, and the robot seeks to move through the free space of the environment to the location that maximizes the occupancy cost associated with the visitor map, and minimizes the cost associated with the robot location map. Intermittently here is on the order of once every 30 minutes.

Path planning once the intended destination location has been determined is accomplished using traditional path planning techniques (e.g., a greedy search[11]) based on a discredited map of the gallery free space. The robot then executes this planned motion, and once it has reached its intended destination the “activity map” is reset and the map of already visited locations is updated, and the process repeats.

A. Motion base

As originally built in the 1990’s, the TRC LabMate[10] was a battery-powered differential drive mobile base with substantive payload. Onboard computation was provided through Motorola 68000 processors, and the robot base interacted with the external world through a RS232 link. Although the LabMate power and optical encoder systems have stood the test of time, the onboard electronics had not. These were replaced with a standard 2-axis motor controller which communicates through a USB port to a laptop mounted inside the stone. The laptop communicates through WIFI to offboard computation and sensors. The revised motion base is shown in Figure 2(c). In order to provide support for the entire Sailing Stone, the original LabMate base has been augmented with additional caster wheels and support infrastructure (Figure 2(d)).

B. Visitor localization

The process of identifying locations in the gallery that have been visited by the public recently takes place in two
phases. There is a per-camera process that operates in image coordinates and which estimates differences in the current image from a reference background image. This process is illustrated in Figures 3 and 4. A mask image is used to identify image locations that correspond to views of the floor. The cameras used to view the gallery are placed well above the floor plane and are (more or less) viewing the floor perpendicularly. A second centralized global process integrates the output from the individual per-camera process shown in Figure 5. Each of the cameras is calibrated to a common reference frame enabling visual events to be integrated into a common frame of reference.

The background image is estimated as a per-pixel exponential filter of the raw input image stream. This allows the background to change slowly with changes in gallery lighting. An absolute image difference is computed between the background image and the current input image. The resulting difference map is then converted into image blobs – ideally corresponding to individual visitors to the gallery – which are approximated by a circular region that fits within the blob’s bounding box. Pixel locations that correspond to non-floor pixels are ignored in this process. Blob centroids and circle radius are then communicated from this camera for more central image processing. Figure 4 shows the images that make up the image difference pipeline.

The ground floor of the gallery is assumed to be flat. This enables the views of the floor to be integrated into a single coherent space by estimating the Homography between each cameras’ view of the floor and some global coordinate frame. Data from the various cameras that monitor the gallery space are integrated together to produce a common view of the gallery space. Figure 5 shows a sample merging of the various floor views from the camera network. (Note that it is rarely necessary to compute this image, although it is possible to do so.) In this image, regions that are outside of the view of the camera network or that do not correspond to floor regions are highlighted.

The visitor map is constructed by combining the circular blob centroids from each of the cameras using the image Homographies to bring all of the blobs into a common reference frame. Blobs obtained within a specified temporal window are used to construct a visitor map by summing the circular regions from the camera network over the temporal window. A sample visitor map is shown in Figure 5 superimposed over the integrated view of the exhibit hall floor. This map is constructed by summing over the circular blobs as projected into the global map.

Path planning takes advantage of the relatively simple structure of the free space of the gallery to use a greedy heuristic to plan the path A sample planned motion of the robot u is shown in Figure 6 shows the planning process. Figure 6(a) shows the output of the path planner as a motion plan is identified that moves the robot around the table in the middle of the gallery space. Once the plan has been identified the robot position map is updated to re-weight the robot’s goal map so that this location is less likely to be chosen in the future. The re-weighting function after one motion is shown in Figure 6(b). Here brighter values correspond to heavier weighting against this location as a future goal location.

C. Localizing the sailing stone

The sailing stone is localized through the same camera network that is used to identify visitor activity. The localization process involves two stages; first, to identify potential locations of the Sailing Stone in the gallery; and second, to utilize temporal coherence to identify the true stone location and track it within the gallery space.

1) Potential stone locations: As the Sailing Stone is a natural stone it is actually quite difficult to identify it in a given image as the floor of the test environment is made of similar material. A Random Forest Classifier[12] is used to identify pixels in the images that might correspond to the Sailing Stone. Images of the stone in the test environment are captured and these along with manually segmented image masks are used to train the classifier. Although this process is quite successful in identifying potential pixels in an image that belong to the Sailing Stone, the classification process is far from perfect. To see this, consider the process of stone detection shown in Figure 7. Figure 7(a) shows the stone
operating in Vari Hall at York University. Figure 7(b) shows the raw results from the Random Forest pixel classifier. As can be seen from the figure, although many pixels on the stone have been classified properly, there are holes in the stone image and a number of floor pixels have been improperly classified as belonging to the Sailing Stone. In order to deal with this the classified image is then eroded and dilated in order to reduce the number of isolated pixel locations that have been identified as belonging to the stone. The result of this process is shown in Figure 7(c). A blob detection process is then applied. Blobs that are too small or too large are discarded. For the test image shown here only one blob was detected (identified by a large red circle). Each possible blob detection location is then converted into the global coordinate frame and passed on to the tracking step.

2) Tracking step: A sampling importance resampling filter is used within a Monte-Carlo localization process[13] to track the state of the Sailing Stone. The initial state of the stone is assumed to be known. Commanded stone motions are used to update the estimate of the Sailing Stone’s position. For each particle a new pose is then sampled from a Gaussian distribution describing the error model of the robot’s odometry. The weight of the each particle is adjusted based on the distance from the pose associated with the particle and the position indicated by the nearest blob detected in the image classifier process. Particles in the filter are resampled based on the “effective number”[14]. This threshold helps to prevent resampling when the set of particles seems to represent the true posterior.

V. DISCUSSION AND ONGOING WORK

Given the ability to estimate a map of the gallery of the motion of visitors, and knowledge of where the robot has been in the environment, it becomes possible to estimate where best to move the Sailing Stone so that it interrupts visitors to the space. Operationally, the Sailing Stone exists in one of two modes. The first is a passive mode in which the camera network estimates motion of visitors to the gallery space. After this phase – which lasts around 30 minutes – the robot enters its motion phase during which it moves to the identified “most disruptive location”. This process is then repeated while the gallery is open.

The Sailing Stone project leverages off the shelf security (nanny) cameras to monitor the environment to both determine where to move to and also to estimate the robot’s state within the environment. When it is moving, the Sailing Stone moves very slowly, so the normal process of monitoring the space along the robot’s path for potential collisions is not performed currently. As the robot lacks onboard sensors, the process of determining if the space in front of the robot is vacant would have to be accomplished using the sensor network itself. Perhaps the simplest solution here would be to use the blob detection process to determine if there is a camera that identifies a person in the path of the robot.

It is anticipated that the Sailing Stone project will engage in its first major showing later this year.
Figure 7. Tracking the Sailing Stone. (a) Shows the Sailing Stone operating in Vari Hall at York University. (b) Shows the raw results of the Random Forest pixel classifier. (c) Shows the eroded/dilated version of (b) with identified blobs of the appropriate size indicated by a red circle. Only one such blob was detected in the image. The location of detected blobs is passed on to the localization step.

ACKNOWLEDGMENTS

The support of NSERC Canada and the NSERC Canadian Field Robotics Network (NCFRN) and the MITACS Globalink programs are gratefully acknowledged.

REFERENCES


