The Potential of Affordance Wayfields

EECS 692 Advanced AI

Emily Sheetz

April 23, 2019

1 Introduction

In order for robots to be used as assistive technology, robot manipulation must move beyond pick-and-place into more complicated manipulation tasks. Object affordances, or action possibilities on objects, show promise for expanding the capabilities of robotic manipulation. Representations of object affordances vary depending on the tasks being performed and the intended application. Exploring these representations and determining a general purpose representation for object affordances could provide a standard model and common terminology to direct further research, allowing robots to perform more complicated tasks and be more useful in everyday life.

This paper describes the replication of "Affordance Wayfields for Task and Motion Planning" by Troy McMahon, Odest Chadwicke Jenkins, and Nancy Amato [5]. The selected work explores representing object affordances as affordance wayfields. This representation allows for planning at any point in configuration space while performing the action. The selected work demonstrates the affordance wayfield formulation on table assembly and toolbox tasks, which involve manipulation actions beyond pick-and-place. The replication efforts concretely define affordance wayfields through potential functions and explore working with screwdrivers and drawers in simulation. Despite challenges with reimplementing the presented work, the replication project demonstrates the promise of affordance wayfields as a representation for object affordances in manipulation tasks.

The paper is outlined as followed: Section 2 describes the work to be replicated in detail; Section 3 reviews related work to provide context for the replicated paper and replication work; Section 4 describes the process of replicating the work; Section 5 details the experiments run to evaluate the replication; Section 6 addresses extensions to the replicated work; Section 7 discusses the project as a whole, highlighting key points of the replication work and lessons learned from replication; Section 8 connects the replicated work to my current research interests; and Section 9 summarizes the project as a whole.

2 Work to Replicate

This paper replicates "Affordance Wayfields for Task and Motion Planning" by Troy McMahon, Odest Chadwicke Jenkins, and Nancy Amato [5]. This paper presents *affordance wayfields* as a representation for object affordances. Affordance wayfields are defined as "an objective function mapped over configuration space" and more specifically, "a set of critical regions that the robot must pass through to perform a task." The main idea is that affordance wayfields attract the robot to configurations that will lead to a completed task and repel the robot away from configurations that collide with obstacles. The representation of the regions are intentionally ambiguous, with the only requirement being that there is some way to determine if a node in the path is in a particular region. Previous representations of object affordances break a particular action down into a series of waypoints the robot end-effector must pass through to complete the action. Instead, affordance wayfields provide the opportunity for planning at every point in configuration space, thereby combining task and motion planning.

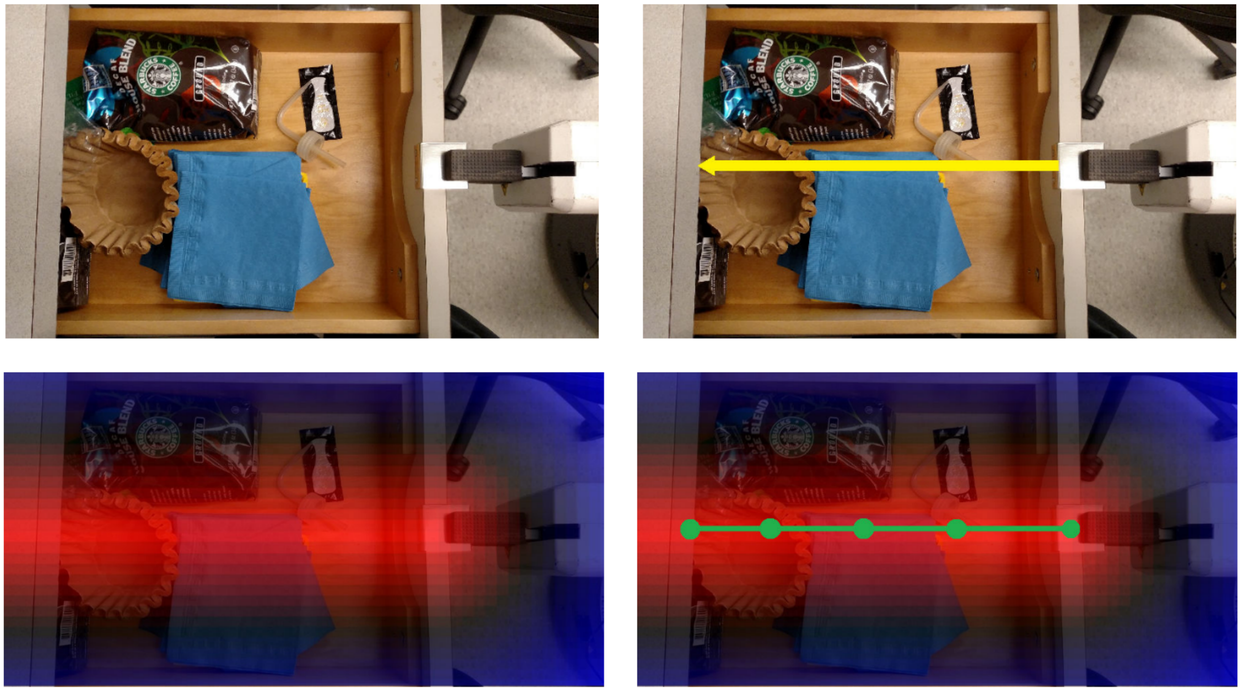

Figure 1: Illustration of affordance wayfields from selected paper [5]. In the upper left image, the robot perceives an open drawer and decides to perform the **close drawer** action. As depicted in the upper right image, the robot must traverse the path indicated by the yellow arrow to perform the **close drawer** action. The lower left image depicts the wayfield matched with drawer. The lower right image depicts the path the robot end-effector must traverse in green.

Figure 1 illustrates how the robot can use affordance wayfields to perform the **close drawer** action. In order to close the drawer, the robot must follow a path along the drawer's axis of motion. By matching the **close drawer** wayfield to the currently open drawer, we can see that the robot must descend and move along the low cost red nodes in configuration space. Following the trajectory along this basin of low cost configurations will complete the **close drawer** action.

To travel through the set of ordered regions in configuration space that define a particular affordance wayfield, the paper uses a modified gradient descent algorithm that minimizes the total cost of the path through the wayfield. The cost of a node in the path is defined by the obstacle cost—the distance required to move the robot end-effector to a free configuration—transitional cost—the cost of transitioning between nodes in the path—and wayfield cost—the cost provided by the wayfield indicating whether the configuration is desirable in the context of the task being executed. The paper proposes what they call an approximate gradient descent algorithm. As opposed to traditional gradient descent formulations that evaluate nodes based solely on the predecessor and successor nodes in the path, the proposed approximate gradient descent algorithm samples nodes near the node being evaluated and moves the node to give the lowest path cost.

To test the affordance wayfield formulation, the paper describes the results of experiments executed on the Laboratory for Progress Fetch robot performing table assembly and toolbox tasks. The table assembly task involves grasping the table leg, lifting the table leg, positioning the table leg, inserting the table leg into its proper place in the table, and screwing the table leg into the table. We can see the robot performing the final few steps of this process in Figure 2. To perform this task, the paper defines a **grasp** wayfield, **position** wayfield, and **screw** wayfield. The **grasp** and **position** wayfields attract the robot end-effector to predefined positions that satisfy the conditions of the table assembly task. The **screw** wayfield affects only the wrist joint of the robot, resulting in a screwing motion. By applying each of these affordance wayfields

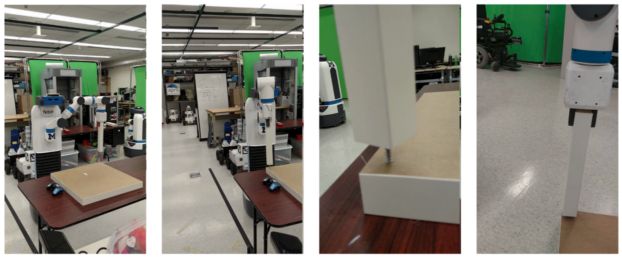

Figure 2: Execution of part of the table assembly experiment [5]. From left to right, we see the robot lifting the table leg, positioning the table leg over the hole, inserting the table leg into the hole, and screwing the table leg into place.

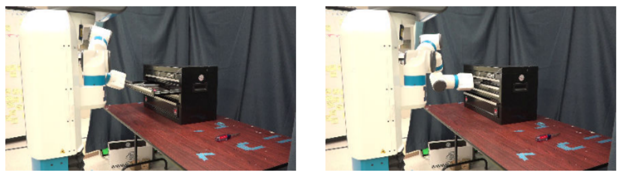

Figure 3: Execution of part of the toolbox experiment [5]. We can see the robot closing the toolbox drawer and arriving at the goal state, with the tools put away and the drawer closed.

in turn, the robot is able to complete the table assembly task.

The affordance wayfield formulation is also tested in a toolbox task. The toolbox task involves grasping and opening the drawer of the toolbox; picking, lifting, and placing a screwdriver and hammer into the drawer; and closing the drawer of the toolbox. We can see the robot reaching the goal state of this experiment in Figure 3. Performing this task involves defining wayfields for **grasp**, **pick**, **place**, **open drawer**, and **close drawer**. The **grasp**, **pick**, and **place** wayfields all attract the robot end-effector to predefined positions to perform the required actions. The **open drawer** and **close drawer** affordances similarly attract the robot end-effector to the drawer handle and towards the predefined location that indicates the drawer is either open or closed.

The affordance wayfield formulation shows promise in that it provides an explainable representation of object affordances while providing opportunity for planning throughout the trajectory of an action rather than just at waypoints. I wanted to work with this idea because it explores actions beyond pick-and-place and provides an intuitive explanation of how to perform a particular action. Figure 1 particularly caught my attention. Performing an **open drawer** or **close drawer** task involves understanding how the object being acted on moves. Furthermore, this illustration of object affordances as affordance wayfields provides a clear, intuitive explanation for what the robot must do to perform the action. We can clearly see the trajectory the robot must follow by visualizing the wayfield cost of each possible configuration in configuration space. These key ideas of explainability of robot action and moving beyond pick-and-place made this paper stand out.

However, the paper has clear weaknesses. Though they can be reused between tasks—for example, both the table assembly and toolbox task involve **grasp** wayfields—affordance wayfields must be manually generated. They also must be manually aligned with the object to be acted on, which is a process known as wayfield matching. Automating both the definition of affordance wayfields and wayfield matching are key challenges to overcome for affordance wayfields to be a general purpose representation of object affordances.

For the replication, I wanted to focus on maintaining explainability and exploring the **open drawer**, **close drawer**, and **screw** actions, specifically because these actions move beyond pick-and-place typical of robot manipulation tasks.

3 Related Work

Robotics research is pushing towards completing complex tasks that involve more than pick-and-place actions. Object affordances have received much interest within the field of robotics. Originally proposed by psychologist J. J. Gibson [2], the concept of object affordances represents the agent's perception of its environment in terms of how it interacts with the environment. Within the field of robotics, object affordances represent *action possibilities* on objects.

Different representations of affordances can achieve different goals. Kokic et al. [4] represent affordances as task labels such as **cut** or **pour** in order to generate task-specific robot grasp poses. Zhu et al. [10] use a combined representation of affordances—including a task label, agent pose, and agent-object relative pose—to perform reasoning tasks in a knowledge base. While these representations are sufficient for generating grasps or reasoning about object affordances, manipulation tasks require a richer representation.

The selected affordance wayfield paper [5] is directly influenced by work involving affordance templates and control basis. Hart et al. [3] propose affordance templates, which represent actions in terms of end-effector waypoints. Affordance templates are robust in that parameters can be adjusted to apply the template to novel object instances. However, they only provide control at the waypoints. Affordance wayfields are inspired by affordance templates, but aim to provide control throughout the entire trajectory of the associated action, emphasizing high-level task planning. Affordance wayfields essentially bridge the gap between high-level task planning—for example, the task of closing a drawer—and low-level motion planning—determining the end-effector configurations necessary to traverse a satisfying trajectory.

Platt et al. [7, 6] propose the control basis for manipulation, which formulates manipulation tasks as interacting potential functions. For example, moving the robot end-effector while grasping an object can be formulated as a **motion** potential function and a **grasp** potential function. In order to move without releasing the grasp on the object, we say that the motion is *subject to* the grasp. The error between the goal configuration and the current configuration is then a sum of the grasp error and the motion error projected into the nullspace of the grasp. The nullspace projection mathematically represents the *subject to* relation by ensuring that any change in configuration that satisfies the **motion** potential does not disturb the progress made towards satisfying the **grasp** potential. Affordance wayfields are similar to control basis by formulating object affordances in terms of the gradient that will generate motions that direct the configuration to satisfying conditions.

Placed in the context of previous work, affordance wayfields generally improve upon previous representations of object affordances by providing a more general purpose candidate for a standard model. The relation of affordance wayfields to affordance templates [3], control basis [7], and nullspace projection [6, 8] provide the theoretical foundation for the replication efforts.

4 Replication Efforts

The replication efforts can be defined in three phases. Each phase culminated in a realization that affected the direction of the replication and a lesson about replicating research. The phases of the replication effort are described in the sections below, while the corresponding lessons learned will be discussed in Section 7.

4.1 Code from Replicated Work

Since my advisor, Chad Jenkins, supervised the affordance wayfield project, I was able to find the code in our lab repository. Looking through this code, there seemed to be distinct differences between the work presented in the paper and how the code implementing these experiments actually worked. These differences forced me to make a decision about how the work would be replicated and what key ideas of the work should be preserved during replication.

Experiments in the paper utilized a variety of affordances, such as **grasp**, **pick**, **place**, **open/close drawer**, **position**, and **screw**. The code only seemed to support a **grasp** and a **rotate** action. The gradients that directed the motion of the actions were categorized as either square or circle motions. The combination of **grasp** and **rotate** actions with square or circle gradients did not seems sufficient to reproduce the experiments described in the paper, nor did they support the variety of actions that any representation of object affordances should support.

The paper placed emphasis on the need to use the proposed approximate gradient descent algorithm. Instead, the code uses cyclic coordinate descent (CCD), which is a type of gradient descent used for inverse kinematics of robots [9].

Most importantly, the code seemed to obfuscate the representation of affordance wayfields and the methods for performing the table assembly and toolbox tasks. Any notion of explainability seemed to be lost looking at the code. Because of this, a challenge of the project was to replicate the work while maintaining the explainability of the paper. My goal was to support diverse actions in an explainable way, possibly through visualizations similar to Figure 1.

Because of the differences between the code and the paper—specifically with respect to actions, motions, algorithm, and explainability—the code for the affordance wayfield project was not used as a reference during the replication.

4.2 Affordance Wayfields as Regions of Configuration Space

Without any code to look at as a reference, the next phase of replication involved defining affordance wayfields as they were described in the paper. The first attempt at defining wayfields focused on the paper's description of affordance wayfields as regions of configuration space.

To begin, I wanted to visualize what motions would look like in 3D space. Figure 4 shows what I call *canonical wayfields*, which are simply visualizations of how particular actions would look in 3D space. For example, in order to grasp an object, the end-effector might start at high cost positions far away from the desired object. The canonical grasp affordance wayfield shows that the end-effector would descend along a conical shape to the low cost positions representing the region of space surrounding the graspable object. In order to open or close a drawer, the end-effector might start away from the drawer's axis of motion. It would descend down into the basin of low cost positions and would then move freely along the drawer's axis of motion. The purpose of the canonical wayfields is to start to represent how the robot end-effector should move through a region.

While the canonical affordance wayfields provide a visual way to explain an action, inspired by Figure 1, they cannot be used to carry out actions. In order to support high-level task planning, the wayfield needs to define a cost for every end-effector location in the area surrounding the object being acted on. The need to define a cost over every end-effector location led to *dense affordance wayfields*. Figure 5 shows the representation of affordance wayfields as dense regions of 3D space. Suppose we have some cube of 3D space we want the end-effector to move through to achieve a particular action. We can attach this cubical wayfield to the object being acted on, and then from any point in the space, gradient descent would direct the end-effector towards the desired object or desired position in the space.

While dense affordance wayfields also provide a visualization for how the end-effector should move through space and represent wayfields as regions of 3D space, they are impractical for action execution. The dense wayfields in Figure 5 only show the cost for every end-effector position in Cartesian space, but end-effector positions also include rotational components. This means that to define a dense wayfield over N points in configuration space would require N^6 explicit cost values, for each of the 6 degrees of freedom at each point. This is intractable and requires too much memory. Furthermore, it makes manually defining affordance wayfields more complicated by explicitly considering every individual end-effector position.

The problem here is that the affordance wayfield paper intentionally left the representation of the wayfields ambiguous. While the idea of affordance wayfields seemed explainable and clear upon first reading the paper, I realized it lacked the sound theoretical foundation required to implement the work. The ambiguous description of what an affordance wayfield actually is forced me to flounder for a concrete representation. While the canonical and dense wayfields provide helpful visualizations of affordance wayfields, they are impractical for action execution and manipulation tasks. Focusing too much on thinking about wayfields as regions of configuration space and maintaining the visual explainability led me down a path that did not help

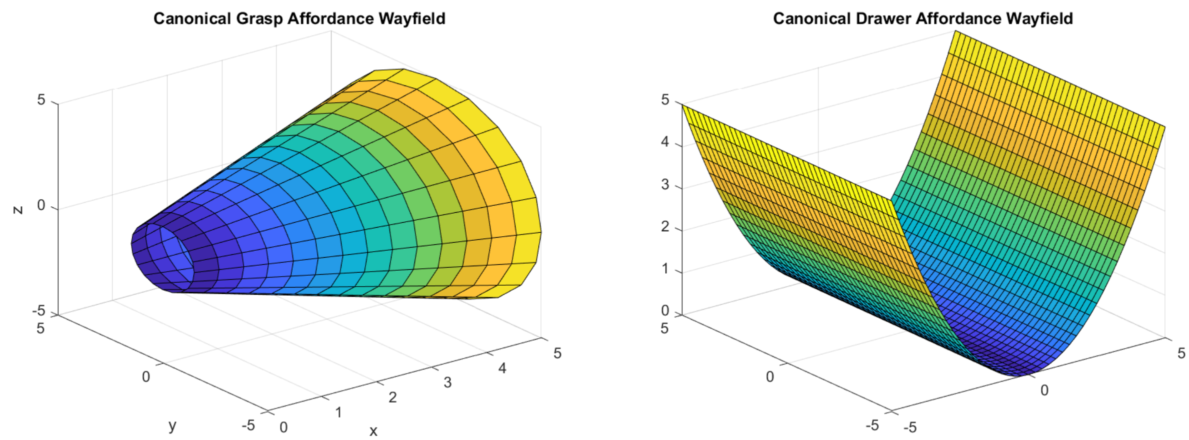

Figure 4: Canonical affordance wayfields for **grasp** and **open/close drawer** actions. These representations are for visualization purposes, to recreate the explainable image in Figure 1. These images show the cost of end-effector positions, where yellow indicate high cost positions and purple indicate low cost positions. These figures were created using MATLAB.

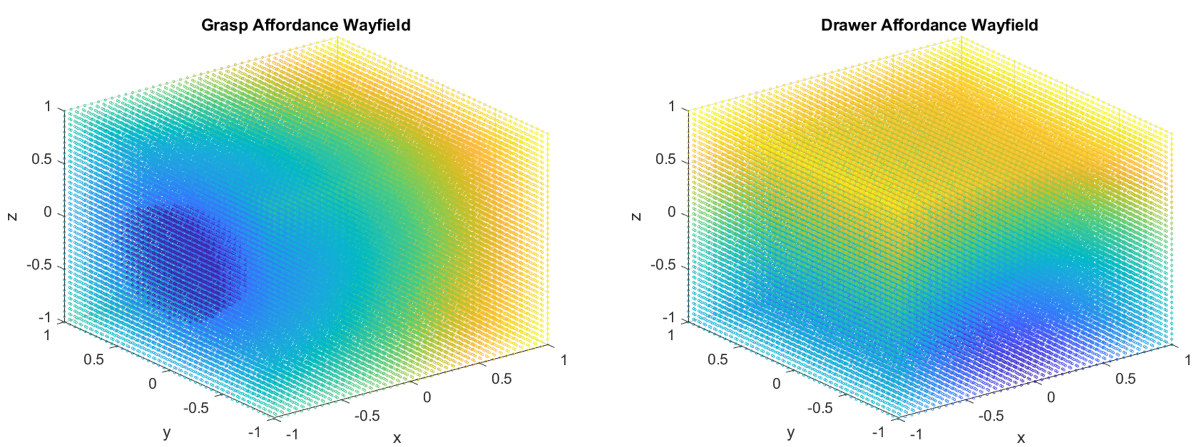

Figure 5: Dense affordance wayfields for **grasp** and **open/close drawer** actions. These representations evaluate the cost of every end-effector position surrounding the object being acted on. These images show the cost of end-effector positions, where yellow indicate high cost positions and purple indicate low cost positions. These figures were created using MATLAB.

in terms of actually implementing the work. The next challenge of the project was to find a solid theoretical foundation in which to ground the work.

4.3 Affordance Wayfields as Potential Functions

Upon rereading the affordance wayfield paper, the relationship between affordance wayfields and control basis seemed more relevant. This directed the replication effort towards thinking of affordance wayfields as an objective function mapped over configuration space, where the functions provide a simpler, more concrete, and more practical representation of affordance wayfields than the canonical and dense affordance wayfield representations.

Affordance wayfields generally direct the position $\boldsymbol{x}$ of the end-effector to a desired location associated with executing a particular action. This is achieved by changing the configuration $\boldsymbol{q}$ of the robot joints. The problem of inverse kinematics considers how to put the end-effector in a particular location [1]. Solving inverse kinematics means relating a desired change in end-effector position $\frac{\partial \boldsymbol{x}}{\partial t}$ to the change in configuration $\frac{\partial \boldsymbol{q}}{\partial t}$ that will direct the end-effector towards the desired location. These quantities can be related by the pseudoinverse—denoted $(\cdot)^+$—of the manipulator Jacobian $\boldsymbol{J}$:

$$\frac{\partial \boldsymbol{q}}{\partial t} = \boldsymbol{J}^+ \frac{\partial \boldsymbol{x}}{\partial t} \tag{1}$$

The manipulator Jacobian $\boldsymbol{J}$ represents the change each joint in the kinematic chain can make along its axis of motion to push the current end-effector location toward the desired location. Inverse kinematics is essentially gradient descent, where we want to minimize the error between the goal end-effector location and the current end-effector location.

Control basis [7] represents fundamental actions as potential functions. Each potential ϕ_i minimizes its associated error ε_i by taking steps in the direction of the gradient $\frac{\partial \varepsilon_i}{\partial \boldsymbol{q}}$. In this case, the gradient represents the error with respect to the configuration $\boldsymbol{q}$ of the robot joints.

Potentials can be composed using the *subject to* relation. For two potentials ϕ_i and ϕ_j, we can say "potential j is subject to potential i," which is denoted:

$$\phi_j \lhd \phi_i \tag{2}$$

This means that we want to achieve the objective of potential ϕ_j with respect to the objective of potential ϕ_i, or that any motion taken to satisfy ϕ_j should not disturb ϕ_i.

In order to carry out some action, we need to compute the gradient with respect to the configuration of the robot. This will tell us how the current configuration needs to change to direct us to the desired configuration associated with the executed action. Actions can be thought of as interacting potentials. Computing the total gradient from the gradients of individual potentials involves projection into the nullspace of a potential. More explicitly, let $\phi_j \lhd \phi_i$ and let $\mathcal{N}\left(\frac{\partial \varepsilon_i}{\partial \boldsymbol{q}_i}\right)$ be the nullspace of the gradient of potential ϕ_i. The total gradient used to direct the robot towards a satisfying configuration can be computed:

$$\frac{\partial \varepsilon}{\partial \boldsymbol{q}} = \frac{\partial \varepsilon_i}{\partial \boldsymbol{q}} + \mathcal{N}\left(\frac{\partial \varepsilon_i}{\partial \boldsymbol{q}_i}\right) \frac{\partial \varepsilon_j}{\partial \boldsymbol{q}} \tag{3}$$

The second term in Equation 3 projects the gradient of potential ϕ_j into the nullspace $\mathcal{N}$ of potential ϕ_i. This means that potential ϕ_j will operate with respect to potential ϕ_i and without disrupting the progress made toward satisfying ϕ_i.

Platt et al. explore composing potentials through nullspace projections [6]. This paper presents the computation of the nullspace:

$$\mathcal{N}\left(\frac{\partial \varepsilon_i}{\partial \boldsymbol{q}}\right) = I - \frac{\partial \varepsilon_i}{\partial \boldsymbol{q}} \frac{\partial \varepsilon_i}{\partial \boldsymbol{q}}^+ \tag{4}$$

For the purposes of this project, we can think about changes in configuration rather than using error gradients to induce the desired change in end-effector location. Instead of computing an error gradient as in Equation 3, we can think about change in configuration as in Equation 1. In this formulation, we can compute the nullspace of manipulator configuration:

$$\mathcal{N} = \boldsymbol{I} - \boldsymbol{J}^{+}\boldsymbol{J} \tag{5}$$

Based on this equation, we can reparameterize Equation 3 in order to compute the total change in configuration:

$$\frac{\partial \boldsymbol{q}}{\partial t} = \frac{\partial \boldsymbol{q}_i}{\partial t} + (\boldsymbol{I} - \boldsymbol{J}^{+}\boldsymbol{J})\frac{\partial \boldsymbol{q}_j}{\partial t} \tag{6}$$

This equation essentially allows the change in configuration induced by potential ϕ_i to dominate, while making the change in configuration induced by potential ϕ_j operate with respect to the dominant objective.

For this project, we will consider a **grasp** potential ϕ_g, **motion** potential ϕ_m, **axis** potential ϕ_u, and **screw** potential ϕ_s. In order to utilize these potentials, we need to define the error for each potential. Initially, we can define the following quantities:

- $\boldsymbol{w}$, the wrench applied to an object by the end-effector.
- $\boldsymbol{x}_{curr}$, the current position of the end-effector.
- $\boldsymbol{x}_{goal}$, the goal position of the end-effector.
- $\boldsymbol{u}$, the axis of motion for a particular object.
- $proj_{\boldsymbol{u}}\boldsymbol{x}$, the orthogonal projection of an end-effector position $\boldsymbol{x}$ onto a particular axis $\boldsymbol{u}$.

From these quantities, we can define the error for each potential:

$$\begin{aligned} \varepsilon_g &= \boldsymbol{w}^T\boldsymbol{w} \\ \varepsilon_m &= \boldsymbol{x}_{goal} - \boldsymbol{x}_{curr} \\ \varepsilon_u &= proj_{\boldsymbol{u}}\boldsymbol{x}_{curr} - \boldsymbol{x}_{curr} \end{aligned} \tag{7}$$

The **grasp** potential error ε_g is zero when the net force applied to the object by the end-effector is zero. The **motion** potential error ε_m directs the end-effector towards a particular goal location. The **axis** potential error ε_u moves the end-effector along a particular axis by minimizing the error between the end-effector position and its projection onto the desired axis of motion. The **motion** and **axis** potential errors are computed with respect to end-effector location $\boldsymbol{x}_{curr}$. As stated in Equation 1, we can compute the required change in configuration using the manipulator Jacobian:

$$\begin{aligned} \tfrac{\partial \boldsymbol{q}_m}{\partial t} &= \boldsymbol{J}^{+}\varepsilon_m \\ \tfrac{\partial \boldsymbol{q}_u}{\partial t} &= \boldsymbol{J}^{+}\varepsilon_u \end{aligned} \tag{8}$$

The **screw** potential ϕ_s does not have a clear error, but it does induce a particular change in configuration:

$$\frac{\partial \boldsymbol{q}_s}{\partial t}_{wrist\_roll\_joint} = \Delta\theta \tag{9}$$

where $\Delta\theta$ is some constant change in angle applied to the wrist roll joint of the robot to induce the screwing motion.

These potentials must interact in order to achieve particular tasks. We can define the relation between potentials:

$$\phi_u \lhd \phi_m, \phi_s \lhd \phi_g \tag{10}$$

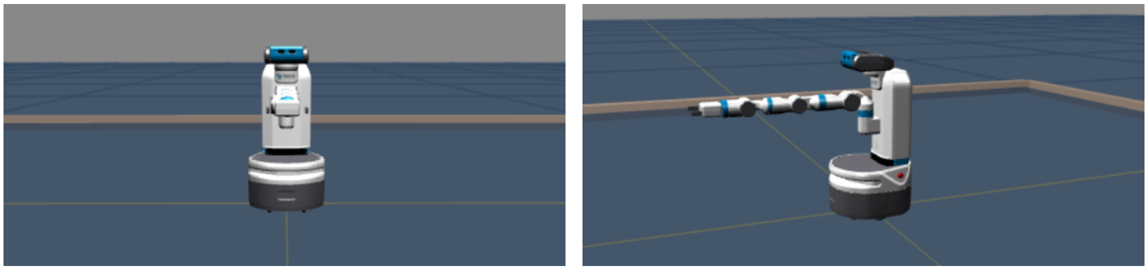

Figure 6: Depiction of KinEval simulation environment with the Fetch robot.

This relation means that maintaining grasp on an object takes highest priority and any other manipulation potential must operate subject to the fact that the object is grasped. The **motion** and **screw** potentials share priority because we want to specify where the end-effector is moving to and how it is moving. The **axis** potential operates subject to all other potentials because it places additional constraints on the other potentials.

Because potential functions often depend on a goal end-effector location, we can still think about affordance wayfields as regions in configuration space anchored at the goal locations. Furthermore, potential functions can be evaluated at any point in configuration space, meaning they satisfy the defining characteristics of affordance wayfields. Formulating affordance wayfields as potential functions provides the theoretical foundation necessary to move forward with replicating the experiments of the selected paper.

5 Replication Experiments

The goal of my replication was to recreate key components of the experiments performed in the selected paper, specifically the **screw** affordance used in the table assembly task and the **open/close drawer** affordance used in the toolbox task. I performed experiments in simulation using the KinEval (Kinematic Evaluator) web-based interface used for class EECS-567 Robot Kinematics and Dynamics. This involved familiarity with HTML and writing code in JavaScript. Though experiments were not tested on the physical robot, translating code into C++ and writing ROS nodes to support operation is a next step for the project. Testing in simulation avoided complications associated with working with hardware.

Figure 6 shows the KinEval simulation environment with the Fetch robot. Object geometries for a screwdriver[1] and drawer[2] were downloaded through free online resources. Since the simulation interface does not provide functionality for determining the amount of force applied by the joints, the **grasp** potential is not considered. Instead, each experiment is initialized so the robot is grasping the object it acts on. This allows the experiments to focus on performing the actual actions associated with each object rather than how to hold the object. The results of each experiment are described in the following sections.

5.1 Screwdriver Experiment

The first replicated experiment involves using a screwdriver and the **screw** affordance. Using a screwdriver involves the **screw** potential, the **motion** potential which directs the end-effector towards the desired location at which we can assume we have completed the task, and the **axis** potential which directs end-effector along the screwdriver's axis of motion. The goal location and axis of motion utilized by the **motion** and **axis** potentials are hard-coded features of the screwdriver object. The necessary potentials interact in the following way:

[1]https://www.turbosquid.com/
[2]https://www.cgtrader.com/

Figure 7: Experimental setup for the screwdriver task. The robot is initialized to this pose, which is similar to the initial pose of other experiments done in the Laboratory for Progress. The robot starts with the screwdriver grasped by the gripper.

$$\phi_u \lhd \phi_m, \phi_s \tag{11}$$

At any time t, the change in configuration $\boldsymbol{q}$ that directs the robot towards a configuration satisfying the conditions of the screwdriver task is:

$$\frac{\partial \boldsymbol{q}}{\partial t} = \frac{\partial \boldsymbol{q}_s}{\partial t} + \frac{\partial \boldsymbol{q}_m}{\partial t} + (\boldsymbol{I} - \boldsymbol{J}^+ \boldsymbol{J}) \frac{\partial \boldsymbol{q}_u}{\partial t} \tag{12}$$

Recall that the change in configuration caused by the **screw** potential, as stated in Equation 9 only affects the wrist roll joint of the robot, and therefore does not need to be projected into a nullspace.

Figure 7 shows the initial configuration of the robot for the screwdriver task. The robot is grasping the screwdriver and its configuration is based on the initial configuration used by other Laboratory for Progress manipulation tasks. When task execution begins, the robot uses inverse kinematics as gradient descent to approach the desired configurations for task completion.

A video of the simulated screwdriver task is available online.[3] In the video, we can see initial instability as the robot begins descending towards the goal location. This is likely due to the fact that I implemented my own inverse kinematics functions. A more widely used inverse kinematics library would not have these issues. Figure 8 shows the final state of the robot after executing the screwdriver task. The end-effector performed the screwing motion and reached the desired goal configuration while moving along the screwdriver's axis of motion, which is the vertical axis. However, we can see that the screwdriver is not aligned with the vertical axis. Since the screwdriver is not properly aligned with its axis of motion, the task of screwing in a screw

[3] https://youtu.be/NUq1VZ-mysQ

Figure 8: End state of the screwdriver task. The goal position has been reached by the **motion** potential while performing the **screw** action. The **axis** potential did move the end-effector along the screwdriver's axis of motion, but notice that the screwdriver is not aligned with the vertical axis.

would not have been completed. This means that the **axis** potential is not sufficient for carrying out this task. The **axis** potential needs to be modified to address the orientation of the end-effector.

5.2 Drawer Experiment

The second replicated experiment involves interacting with a drawer to perform the **open drawer** and **close drawer** affordances. These actions involve the **motion** potential which directs the end-effector towards the desired locations at which we can assume the drawer is open/closed and the **axis** potential which directs the end-effector along the drawer's axis of motion. The goal locations and axes of motion utilized by the **motion** and **axis** potentials are hard-coded features of the drawer object. The necessary potentials interact in the following way:

$$\phi_u \lhd \phi_m \tag{13}$$

At any time t, the change in configuration $\boldsymbol{q}$ that directs the robot towards a configuration satisfying the conditions of the drawer task is:

$$\frac{\partial \boldsymbol{q}}{\partial t} = \frac{\partial \boldsymbol{q}_m}{\partial t} + (\boldsymbol{I} - \boldsymbol{J}^{+}\boldsymbol{J})\frac{\partial \boldsymbol{q}_u}{\partial t} \tag{14}$$

Figure 9 shows the initial configuration of the robot for the drawer task. The robot is grasping the handle of the drawer and the initial configuration is initialized somewhat arbitrarily to what seemed like an appropriate grasp configuration. When task execution begins, the robot uses inverse kinematics as gradient descent to approach the desired configurations for task completion.

A video of the simulated drawer task is available online.[4] Notice that the drawer does not actually open and close. This is caused by the nature of the imported drawer geometry. The drawer is considered one solid object rather than an articulated object whose parts move independently. Nevertheless, we can see how the robot would interact with this object if the top drawer followed the end-effector. Figure 10 shows the final state of the robot after executing the drawer task. The end-effector bounces back and forth between the open and close goal locations while moving along the drawer's axis of motion, which is the horizontal

[4] https://youtu.be/0qNEwaf_iBU

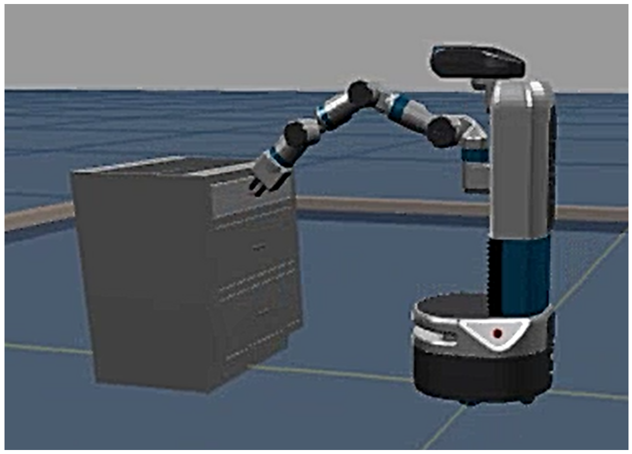

Figure 9: Experimental setup for the drawer task. The robot is initialized to this pose, with the drawer handle grasped by the gripper.

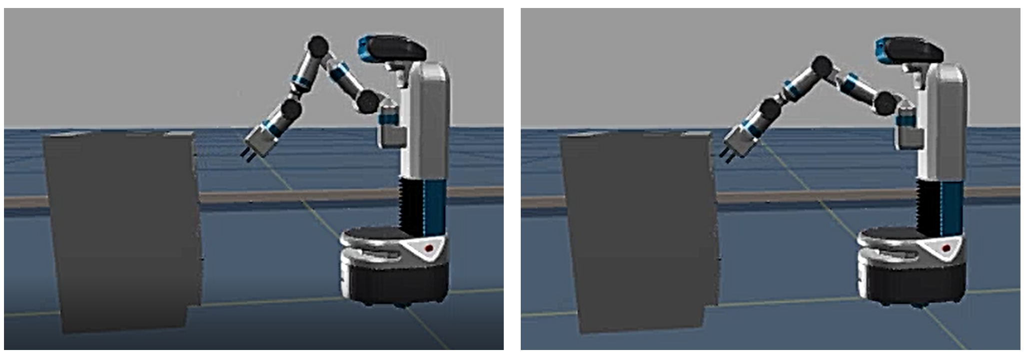

Figure 10: Open and close positions of the drawer task. The **motion** potential and **axis** potential continuously move the end-effector along the drawer's axis of motion between the drawer's open and close goal positions.

axis. Though this experiment works well, the goal location for the **open** affordance is actually very close to the goal location for the **close** affordance. If these goal locations were any farther apart, the implemented inverse kinematics functions would cause much instability and the robot would not complete either the **open drawer** or the **close drawer** actions.

6 Extensions

The replication of this work did not include any experimental extensions, but it did involve theoretical extension of the ambiguous affordance wayfield representation to a more concrete description of affordance wayfields as potential functions. A literal replication of the paper would have kept the representation of affordance wayfields as ambiguous regions of configuration space. However, extending this representation to the more specific composition of potential functions allows for an implementation of the paper that preserves the core ideas of gradient descent using an objective function mapped over all of configuration space, complex manipulation beyond pick-and-place, and explainability. Though potential fields are not easily depicted visually as affordance wayfields are in Figure 1, breaking down a task into smaller fundamental motions and composing them together to allow for completion of a task maintains, in my mind, the explainability that originally made the affordance wayfield paper so interesting. Even though the work was not extended by performing additional experiments, the added step of grounding affordance wayfields in a sound theoretical foundation extends the work by making the idea, theory, and implementation more straightforward.

7 Project Discussion

As part of the replication, I made a point to avoid any hacks to get demonstrations to work and to maintain explainability. To that end, it seems important to acknowledge a few hacks here. The potential relations used to perform the screwdriver and drawer experiments were originally formulated a different way, but were reformulated as seen in Equations 11 and 13 after several experiments would not work at all. The positions that indicate whether the drawer is open or closed are very close together in the drawer experiment. If these goal locations were any farther apart, the robot would quickly get stuck in a particular configuration and not move. The fact that these locations are so close together means we are operating under a loose definition of "open," and "ajar" might be a better descriptor for the action being executed. Finally, the initial configurations of the robot for each experiment as depicted in Figures 7 and 9 may make it easier for the robot to complete the task. Though other initial configurations were not explored, these configurations were not picked specifically to make the experiments work as expected. They were selected somewhat arbitrarily based on conventions in the Laboratory for Progress and how I thought the robot might grasp a drawer.

The replicated experiments involving the screwdriver and the drawer demonstrate that affordance wayfields can be used to complete tasks. Since they can be used to represent a variety of actions, affordance wayfields may provide an option for a general purpose representation of object affordances. Object affordances allow robotics research to move beyond simple pick-and-place tasks into more complex manipulation tasks. My replication efforts provide concrete evidence that affordance wayfields represent object affordances in a way that allow for manipulation beyond pick-and-place. Replication did emphasize the need for further work, specifically the manual application of affordance wayfields. For the experiments, I hard-coded the goal positions and axes of motion for the **screw**, **open drawer**, and **close drawer** actions. Though this was relatively simple, in practice hard-coding will not be scalable. Further research is needed to automate the processes of defining wayfields and wayfield matching. Performing experiments on the physical robot may also provide more insight into the methods.

This replication project taught me several lessons. As described in Section 4.1, looking at the code used to implement this project taught me that sometimes the ideas presented in the paper and the actual implementation details in code differ. This may be caused by pressure to publish or demonstrate that an idea shows promise. Moving forward with my own research, I want to emphasize explainability, openness specifically in terms of what factors may affect the results of my experiments, and not cutting corners to push a paper through to publication.

The challenges that arose from the ambiguity of the paper, described in Section 4.2, taught me that good research requires a strong theoretical foundation. The idea behind the paper caught my attention and convinced me that this work would be straightforward to replicate. However, the moment I sat down to write code, I realized that the idea was nowhere near enough. Only once the idea and theory were in place did the implementation details become clear. For this project, writing code took a matter of hours over the course of a few days while formalizing the problem and exploring the theoretical background took weeks. As I continue with my own research, I will make sure I formulate the problem clearly and specifically, provide sufficient theoretical foundation to ground my ideas, and describe the tools needed to implement the work.

Overall, I learned that replication can be very challenging. But given how important replication is to the integrity of research, my goal is to make replicating my own works as straightforward as possible. Most importantly, I have new appreciation for how ideas, theory, implementation, and experiments relate and depend on each other in order to make a strong research project.

8 Connection to Current Research

I chose to replicate this work on affordance wayfields because it is closely related to my own research. I am working with Professor Jenkins in the Laboratory for Progress. I am interested in how object affordances can be used to improve robot manipulation. I think there should be a standard model of object affordances that is general purpose and rich enough to be used to complete diverse tasks in a variety of applications. Having read the original paper and spent a semester replicating the work, I think affordance wayfields formulated as interacting potential functions provide important information about how to interact with objects and provide an opportunity for exploring representations of object affordances. However, this project supported my hypothesis that affordance wayfields alone are not enough. Manual generation and application of affordance wayfields are not practical if we want the robot to perform high-level tasks autonomously and in a scalable way. Furthermore, affordance wayfields do not represent pre- and post-conditions of actions, which would make manipulating objects in clutter challenging. This project has emphasized the need for further research relating to autonomous definition and matching of wayfields in order to minimize hard-coding. My replication of this research with affordance wayfields will help me define a general representation of object affordances robust enough to be applicable in a variety of manipulation tasks.

9 Conclusion

For this project, I replicated "Affordance Wayfields for Task and Motion Planning" by Troy McMahon, Odest Chadwicke Jenkins, and Nancy Amato [5]. This paper represents object affordances as affordance wayfields, and demonstrates the push of robotics research towards complex manipulation tasks beyond pick-and-place. Affordance wayfields are directly influenced by previous research related to affordance templates [3]—which provide control during action execution at waypoints—and control basis through potential functions [7] and nullspace composition [6]—which break tasks down into fundamental actions that can be executed concurrently. To compensate for ambiguous definitions in the paper, this project formulates affordance wayfields as interacting potential functions while maintaining the key ideas of the paper, which are gradient descent to generate motions, planning and control at any point in configuration space, and explainability. The replicated work was tested by a screwdriver task and a drawer task, which generally showed the promise of affordance wayfields as a representation of object affordances. The original work on affordance wayfields was extended with a stronger and clearer theoretical foundation by defining affordance wayfields as interacting potential functions.

This replication process taught me the importance of openness, explainability, good ideas, strong theoretical foundations, and clear implementation details. These are lessons I can carry with me into my future research, as I continue to improve upon affordance wayfields and incorporate them into a rich, general purpose representation of object affordances.

References

[1] Howie M Choset, Seth Hutchinson, Kevin M Lynch, George Kantor, Wolfram Burgard, Lydia E Kavraki, and Sebastian Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementation.* MIT press, 2005.

[2] James J. Gibson. The Theory of Affordances, 1977.

[3] Stephen Hart, Paul Dinh, and Kimberly Hambuchen. The Affordance Template ROS Package for Robot Task Programming. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 6227–6234, 2015.

[4] Mia Kokic, Johannes A. Stork, Joshua A. Haustein, and Danica Kragic. Affordance Detection for Task-Specific Grasping Using Deep Learning. *IEEE-RAS International Conference on Humanoid Robots*, pages 91–98, November 2017.

[5] Troy McMahon, Odest Chadwicke Jenkins, and Nancy Amato. Affordance Wayfields for Task and Motion Planning. *IROS*, pages 2955–2962, 2018.

[6] Robert Platt Jr, Andrew H Fagg, and Roderic A Grupen. Nullspace Composition of Control Laws for Grasping. *IEEE International Conference on Intelligent Robots and Systems*, 2002.

[7] Robert Platt Jr, Andrew H Fagg, and Roderic A Grupen. Manipulation Gaits: Sequences of Grasp Control Tasks. 1(April):801–806, 2004.

[8] Robert Platt Jr, Andrew H Fagg, and Roderic A Grupen. Nullspace Grasp Control: Theory and Experiments. *IEEE Transactions on Robotics*, 26(2):282–295, 2010.

[9] L-CT Wang and Chih-Cheng Chen. A combined optimization method for solving the inverse kinematics problems of mechanical manipulators. *IEEE Transactions on Robotics and Automation*, 7(4):489–499, 1991.

[10] Yuke Zhu, Alireza Fathi, and Li Fei-Fei. Reasoning about Object Affordances in a Knowledge Base Representation, 2014.