This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Simulating Polyculture Farming to Learn Automation Policies for Plant Diversity and Precision Irrigation

Yahav Avigal, *Graduate Student Member, IEEE*, William Wong, Mark Presten, Mark Theis, Shrey Aeron, *Student Member, IEEE*, Anna Deza, Satvik Sharma, Rishi Parikh, Sebastian Oehme, Stefano Carpin, *Senior Member, IEEE*, Joshua H. Viers, Stavros Vougioukas, *Senior Member, IEEE*, and Ken Goldberg, *Fellow, IEEE*

***Abstract*— Polyculture farming, where multiple crop species are grown simultaneously, has potential to reduce pesticide and water usage while improving the utilization of soil nutrients. However, it is much harder to automate polyculture than monoculture. To facilitate research, we present AlphaGardenSim, a fast, first order, open-access polyculture farming simulator with single plant growth and irrigation models tuned using real world measurements. AlphaGardenSim can be used for policy learning as it simulates inter-plant dynamics, including light and water competition between plants in close proximity and approximates growth in a real greenhouse garden at 25,000× the speed of natural growth. This paper extends earlier work with a new action space that includes planting, which dynamically finds new seed locations that increases resources utilization, and an adaptive sampling technique to reduce the number of actions taken at each timestep without affecting performance. We also evaluate other automation policies using a novel metric that combines plant diversity and canopy coverage. Code and supplementary material can be found at https://github.com/BerkeleyAutomation/AlphaGarden.**

***Note to Practitioners*—Monoculture farming is often characterized by heavy agrichemical inputs, such as chemical fertilizers and pesticides, and increased vulnerability to disease and pestilence. This paper is motivated by the lack of long-term sustainability of industrial agriculture, and its implications for human food security. Although polyculture is a sustainable alternative to monoculture farming, it requires more human labor and is more challenging to automate. In this paper we propose a fast, first order simulator that simulates the growth of plants in a polyculture setting. Simulation experiments suggest that the simulator can be used to learn a planting, watering and pruning plan a robot can follow to produce maximal yield from a diverse set of plants with limited irrigation, however it has not yet been tested on a physical garden. In future research we will develop a fully automated controller that will operate planting, irrigation and pruning tools in a physical garden over multiple plant growth cycles.**

***Index Terms*— Agriculture, automation, learning control systems, modeling, simulation.**

Manuscript received October 24, 2021; accepted December 16, 2021. This article was recommended for publication by Associate Editor R. Y. Zhong and Editor M. Dotoli upon evaluation of the reviewers' comments. This work was supported in part by the RAPID: Robot-Assisted Precision Irrigation Delivery Project through the NSF National Robotics Initiative under Grant USDA 2017-67021-25925. *(Corresponding author: Yahav Avigal.)*

Yahav Avigal, William Wong, Mark Presten, Mark Theis, Shrey Aeron, Anna Deza, Satvik Sharma, Rishi Parikh, and Ken Goldberg are with the AUTOLAB, UC Berkeley, Berkeley, CA 94720 USA (e-mail: yahav_avigal@berkeley.edu; goldberg@berkeley.edu).

Sebastian Oehme is with the Department of Electrical and Computer Engineering, TU Munich, 80333 München, Germany.

Stefano Carpin is with the School of Engineering, UC Merced, Merced, CA 95343 USA.

Joshua H. Viers is with the Department of Environmental Systems, School of Engineering, UC Merced, Merced, CA 95343 USA.

Stavros Vougioukas is with the Department of Biological and Agricultural Engineering, UC Davis, Davis, CA 95616 USA.

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TASE.2021.3138995.

Digital Object Identifier 10.1109/TASE.2021.3138995

I. Introduction

CULTIVATING plants has been an essential human activity for over 10,000 years. Many factors influence the quality and quantity of yield, such as irrigation, pesticide use, weather conditions, and plant disease. Industrial agriculture aims to maximize yield by growing a single plant species in isolation (monoculture). Polyculture farming, on the other hand, involves growing different crops simultaneously in imitation of the diversity of natural ecosystems, and is a sustainable alternative that uses biodiversity to reduce pesticides, disease, and weeds [1]–[3]. Polyculture is also more practical for confined urban spaces and essential for aesthetic gardens.

With increasing demands for fresh local herbs and produce, polyculture gardens are increasingly in demand. However, polyculture requires more human labor than monoculture to prune and maintain. A robot with a reliable and sustainable control policy has the potential to increase yield and reduce water consumption. Finding an optimal policy is a challenging task. The long time constants for real-world experiments motivates the use of a simulated environment. It is difficult to simulate inter-plant dynamics, including competition for light, water and nutrients.

This paper is a revised and substantially extended version of two earlier conference papers; Avigal *et al.* [4], [5] which explored how to tune growth and irrigation models using real world measurements, model companionship relationships that affect inter-plant dynamics, and learn an automation policy based on demonstrations from a 1-step lookahead policy.

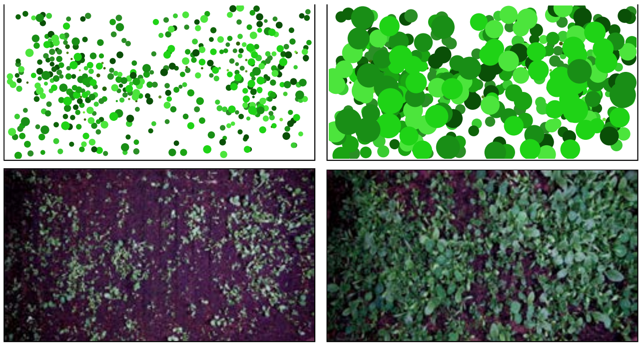


Fig. 1. **Simulated polyculture garden.** Planted with seeds from 13 edible plant types, and grown for a period of 90 days. A garden simulated with AlphaGardenSim, and a corresponding physical garden initialized with similar seed types and locations, and irrigated every 2 days for a similar period of time. Each shade of green in the simulated garden represents a different plant type. **Left:** The gardens at day 15, a few days after germination. **Right:** The gardens at day 30.

Including the contributions from [4], [5], this journal paper makes 5 contributions:

1) A fast, first order simulator, AlphaGardenSim, that incorporates parameterized individual plant growth models, companion plant effects and inter-plant dynamics, tuned with real world measurements from a physical testbed, to simulate competition over resources between plants in close proximity.
2) An updated irrigation model based on the Richards equation, tuned with real world measurements.
3) A new action space that allows an agent to plant seeds dynamically to increase canopy coverage, dynamically adjust plant diversity, and to extend garden lifetime.
4) An adaptive sampling method that reduces the number of actions taken.
5) New simulation experiments evaluating policies using a novel metric that rewards policies for valuing both plant diversity and canopy coverage equally. As a result, policies achieve both high plant diversity and canopy coverage.

II. Related Work

Past work in plant growth simulation has predominantly been focused on monoculture agriculture. Widely used simulation models include DSSAT [6] and AquaCrop [7]. However, such models are intended for simulating large scale, monoculture agricultural operations, and are point-based models, which make the assumption that plants are grown homogeneously. Therefore, these models are not well-suited for a polyculture setting, where gardens are heterogeneous.

GeoSim [8] is a tool that adds spatial functionality to point-based agricultural models by leveraging data from a geographic information system (GIS) to run independent simulations at different geospatial points, allowing for heterogeneous simulation. However, to tune a policy for managing a small-scale polyculture garden, it is desirable to simulate at the individual plant level. Recently, Chebrolu *et al.* [9] developed a point cloud registration algorithm that enables plant monitoring to analyze growth at the single-plant level. It can be used to tune a single-plant growth model, but does not reveal inter-plant interactions, which are required to provide higher granularity data for polyculture modeling.

There exist individual plant models that model inter-plant competition, but to the best of our knowledge, there does not exist one for a polyculture setting. For example, Damgaard *et al.* [10] proposed modeling competition between individual plants based on density and size differences, but their work does not explicitly model resource competition, which is important for tuning a policy that affects the distribution of resources in a garden. Price *et al.* [11] introduced a simulator for individual plant growth and competition with promising results, but their work only modeled plant radii and did not take into account competition for resources other than water. According to Berger *et al.* [12], a review on individual-based approaches for modeling plant competition, existing models lack consideration for the effect of plants on resource levels in an environment. Thus, we were motivated to develop our own first order simulator for tuning a polyculture gardening policy.

Czárán and Bartha [13] proposed a broad classification of individual-based plant competition models as either grid-based models or individual-based neighborhood models. Grid-based models discretize a region into a grid of cells that may be occupied by plants, while individual-based neighborhood models represent plants in a continuous space. Furthermore, grid-based models typically use empirical rules to define plant competition, whereas individual-based neighborhood models define explicit mechanisms that regulate competition. One such individual-based neighborhood model is the zone-of-influence model [12], where a circular zone corresponding to the plant's size defines where a plant acquires resources from. Plants with overlapping zones are in competition with each other, and the growth rate of a plant decreases as more of its zone is overlapped with. While these models allow for greater modeling complexity, grid-based models make simplifying assumptions and reduce computational cost.

Research in agricultural automation has also been conducted specifically in crop modeling and individually controlled plants. Wiggert *et al.* [14] developed a testbed that enables real-time data collection of plant water stress to automate and optimize plant-level irrigation. Habibie *et al.* [15] trained a Simultaneous Localization And Mapping (SLAM) algorithm in simulation to automate fruit harvesting in a red apple tree field. While it supports a wide variety of use cases and enabled successful harvesting, plant dynamics were not modeled, and the simulation focused on a single plant type. CoppeliaSim [16] comes closer to simulating a polyculture garden, as plants are able to be controlled separately. This simulator was used to train a crop monitoring green house robot to navigate a greenhouse and identify diseased crops [17]. Even though each plant had unique parameters, they did not model inter-plant interactions.

III. AlphaGardenSim

In AlphaGardenSim the goal is to grow a lush and diverse polyculture garden, represented as a discrete $H \times W$ grid containing N plants uniformly sampled from a set of k plant

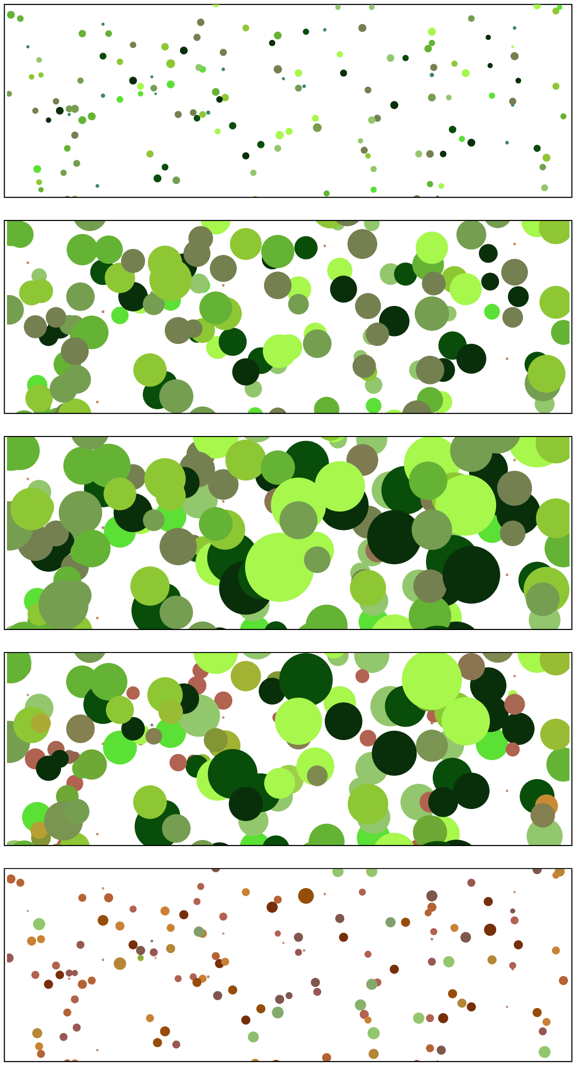


Fig. 2. **Plant Life Stages**. Each plant is modeled with a life cycle trajectory, consisting of five stages (from top to bottom image): germination, vegetative, reproductive, senescence, and death. When plants get underwatered or overwatered, their radius decays exponentially and their color turns brown, and after a short period they move to the death stage. However, if they receive their desired water amount prior to that, they can return to their original stage.

types, as well as types *soil* and *unknown*, within a growing period T while minimizing irrigation. We can frame the general problem as a Partially Observable Markov Decision Process (POMDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{O})$.

A. States ($\mathcal{S}$)

A state $\mathbf{s}(t)$ includes the following quantities at timestep t for every point (x, y) in the garden: the seed locations $\mathbf{c}(x, y)$, the health of each plant $h(x, y, t)$ and the soil moisture levels $w(x, y, t)$ in the garden. The timestep t is in days for AlphaGardenSim. We also introduce a vacancy score $e(x, y, t)$ as the minimum distance from point (x, y) to any plant. We define $\mathbf{d}(x, y, t)$ be a vector of length $k+1$ representing one of the k plant types (or soil) type that is visible overhead at point (x, y). With full state knowledge this is a 1-hot vector, however in a physical garden this induces a distribution over the plant types.

B. Actions ($\mathcal{A}$)

The agent can execute any combination of the following actions, or none, per observation:

- Watering, $a_w(x, y, t)$, applies a fixed amount of water to a circle of radius 9 centered at the center point of the observation (x, y), following the irrigation model described in Section IV-F. The amount of water applied to each grid cell decays exponentially as it approaches the edges of the watering circle.
- Pruning, $a_p(x, y, t)$ reduces the radius of a plant. We define a pruning window of size 5×5, centered at the center point of the observation (x, y). A pruning action will reduce the radii of all plants visible within the pruning window by a pruning level p, which is set by default to $p = 5\%$. This is to simulate the inaccuracy of an automated pruner that is likely to prune plants in the neighborhood of the target leaf.
- Planting, $a_s(x, y, t)$. We extend the action space presented in [4] with a new planting action that seeds a plant at point (x, y) at timestep t. A plant can be planted only in locations labeled as *soil*.

C. Transitions ($\mathcal{T}$)

At each timestep t, AlphaGardenSim executes a sequence of updates across the garden: irrigation, lighting, water use and plant growth according to the models described in Section IV-B.

D. Rewards ($\mathcal{R}$)

As the objective is to achieve a diverse garden with maximal yield and water efficiency, we define $\mathbf{P}(k, t)$, the global population in the garden as a distribution over the k plant types, and the following rewards:

- $r_d(t)$, the garden diversity at timestep t is defined as the normalized entropy of the global population in the garden:

$$r_d(t) = \frac{H(\mathbf{P}(k, t))}{\log k} = \frac{-\sum_{i=1}^{k} \mathbf{P}(i, t) \log \mathbf{P}(i, t)}{\log k}$$

- $r_c(t)$, the garden canopy coverage is defined as the total percent coverage at timestep t, taking into account only the coverage of the plants, ignoring the uncovered space labeled as *soil*:

$$r_c(t) = \frac{\sum_{i=1}^{k} \mathbf{P}(i, t)}{H \cdot W}$$

- $r_w(t)$, the garden water efficiency is defined as the negative water use at day t:

$$r_w(t) = -\sum_{x,y} w(x, y, t)$$

E. Observations ($\mathcal{O}$)

To simulate sensor precision limitations, we define $\mathbf{o}(x, y, t)$, a sector of size $\frac{H}{10} \times \frac{W}{10}$ centered at point (x, y) representing the area observable at timestep t.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

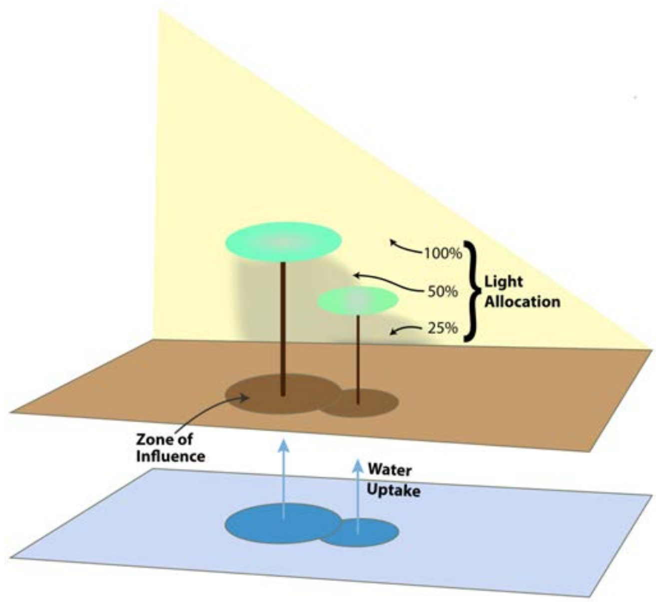


Fig. 3. **Light and Irrigation Models.** Each plant receives light based on the size of its unoccluded leaf area in the grid, i.e., the number of grid points visible overhead, while occluded points allocate light in an exponentially decaying fashion. The plant's water uptake is then drawn from its neighboring grid points, to fulfill its growth potential. The plant is limited by the amount of light it intercepts and the amount of water available in its zone-of-influence.

IV. Modeling

A. Plant Representation

We extend the model proposed by Price *et al.* [11] which represents a plant using a seed location and a radius by adding the height attribute, allowing competition for light in addition to water competition. This abstraction is both efficient and expressive, as it allows to simulate inter-plant occlusions, implicitly leading to competition for resources and complex interactions.

B. Garden Dynamics

Our process-based crop model [4] simulates plant growth according to endogenous plant parameters and environmental conditions. AlphaGardenSim executes a sequence of updates at each timestep: lighting, water use and plant growth.

1) Lighting Update: We assume a fixed light source directly above the garden. To simulate photosynthesis [18] as a part of the plant growth model, plants allocate light based on the size of their leaf area. When a plant is occluded by taller plants, light is distributed in an exponentially decaying fashion, where the i^{th} tallest plant at point (x, y) in the grid receives $(\frac{1}{2})^i$ amount of light from point (x, y), where $i \in \{0, \ldots, n_{xy}\}$ and n_{xy} defines the number of plants for which the distance between their seed location and point (x, y) is smaller than their radius r, as demonstrated in Fig. 3. For plant j with radius r_j, AlphaGardenSim estimates the total amount of light the plant accumulates l_u by a summation over the light allocated from all garden points that are less than distance r_j from the plant's seed location.

2) Water Use: Water uptake is defined by a zone-of-influence model [12], allowing access to soil moisture concurrent to the plant's circular size. In general, allocation of water depends on a plant's allocated light, l_u, and water competition in intersecting coordinates. The allocated light defines the maximal amount of water required by a plant:

$$w_{max} = \frac{c_2}{c_1}\sqrt{l_u}$$

where c_1 and c_2 are plant-specific parameters that control a plant's resource efficiency - c_1 corresponds to water use efficiency and c_2 corresponds to light use efficiency. Larger values for c_1 or c_2 represent higher biomass accumulation per unit of resource at each timestep [19]. In AlphaGardenSim, c_2 is held constant for all plants, thus c_1 can be seen as the biomass accumulation parameter.

Competition for resources occur for each coordinate in the overlapping zones of influence of plants. Available water is randomly distributed among the plants present in the overlapping regions. For each such plant, we allow it to use the maximum amount of water it desires from this coordinate. In intermediate growth stages, this value is defined on a per plant basis as:

$$w_d = \frac{w_{max}}{l_a}$$

where w_d is a plant's desired water amount and l_a is a plant's total leaf area. This approach causes a plant to grow slower in expectation as more of its zone overlaps with the zones of other plants, in accordance with the zone-of-influence model. Furthermore, the water uptake is limited by a soil-specific permanent wilting point that represents a lower bound from which a plant can extract soil moisture [20].

3) Plant Growth: In AlphaGardenSim, we assign each plant values for the growth parameters: germination time, maturation time, growth rate, and growth potential. These values were tuned by monitoring and analyzing the growth of one hundred and twenty real world plants, and averaging the growth of a plant with others of its same species. Growth parameters can be found in Table I. Germination time and maturation time determine a plant's growth stage, as seen in Section IV-D, and are sampled from a normal distribution with a calculated variance. Growth rate, which is defined as the biomass accumulation variable c_1, and growth potential are parameters that directly determine a plant's size.

The amount of allocated light and water resources impact the biomass pool that is available for growth. A plant's growth is modeled as a logistic curve [21]:

$$\tilde{g} = c_1 \cdot \min(w, w_{max}) \cdot (1 - \frac{r_t}{r_1})$$

where w is the actual amount of water this plant was able to adsorb, r_t is the plant's current radius and r_1 is the plant's growth potential, which controls how large the plant will grow.

For each plant, $\tilde{g}$ is then strategically distributed to vertical and radial growth to ensure maximum unoccluded leaf area. Therefore, we define $l_{o,i}$ and $l_{u,i}$, the number of points where plant i is occluded and unoccluded respectively, and model

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE I

GROWTH ANALYSIS: WHERE g_0 (DAYS) IS ORIGINAL GERMINATION TIME, g_1 (DAYS) IS TUNED GERMINATION TIME, m_0 (DAYS) IS ORIGINAL MATURATION TIME, m_1 (DAYS) IS TUNED MATURATION TIME, r_1 IS GROWTH POTENTIAL, c_1 IS THE BIOMASS ACCUMULATION PARAMETER, $c(35)$ (CM^2) IS THE SIMULATED CANOPY COVERAGE ON DAY 35, AND $e(35)$ (cm) IS THE MEAN ABSOLUTE ERROR ON DAY 35 BETWEEN SIMULATED AND AVERAGE REAL WORLD RADIUS. ORIGINAL VALUES WERE TAKEN FROM PUBLISHED PLANT TABLES [26]. GROWTH TIME IS FOUND BY SUBTRACTING g_1 FROM m_1

Plant Type	g_0	g_1	m_0	m_1	r_1	c_1	$c(35)$	$e(35)$
Borage	7	7	49	55	60	0.09	3107	6.61
Kale	3	7	62	55	65	0.10	7450	5.41
Swiss Chard	7	7	53	50	47	0.11	5536	9.93
Turnip	3	7	42	47	53	0.11	3961	10.04
Green Lettuce	7	9	43	52	27	0.08	232	7.46
Arugula	5	8	45	52	40	0.10	1133	5.50
Sorrel	7	15	53	70	8	0.08	59	9.58
Cilantro	7	10	53	65	20	0.09	23	10.76
Red Lettuce	5	12	45	50	28	0.09	10	11.61
Radicchio	5	9	83	55	53	0.09	53	9.28

TABLE II

SIMULATOR STATE, ACTION AND REWARD VARIABLES. DIMENSIONALITY IS SHOWN IN THE SECOND ROW OF EACH VARIABLE SECTION

State Variables				
$\mathbf{d}(x,y,t)$	$h(x,y,t)$	$w(x,y,t)$	$e(x,y,t)$	$\mathbf{c}(x,y)$
$[H,W,k+1]$	$[H,W]$	$[H,W]$	$[H,W]$	$[H,W,k]$
Plant Type	Plant Health	Water Amount	Vacancy	Seed Locations
Action Variables				
$a_p(x,y,t)$		$a_w(x,y,t)$		$a_s(x,y,t)$
$[2,N]$		$[2,N]$		$[H,W,k]$
Pruning		Watering		Planting
Reward Variables				
$r_d(t)$		$r_w(t)$		$r_c(t)$
$[1,]$		$[1,]$		$[1,]$
Plant Diversity		Water Efficiency		Canopy Coverage

this dynamic as follows:

$$l_p = \frac{l_{u,i}}{l_{u,i} + l_{o,i}}$$
$$r_i = \max(k_1, \min(l_p, k_2))$$
$$\tilde{g}_{radial} = r_i \tilde{g}$$
$$\tilde{g}_{vertical} = (1 - r_i)\tilde{g}$$

Here, k_1 and k_2 are plant-specific parameters that control the ratio of $\tilde{g}$ a plant apportions to radial growth - k_1 is the lower bound and k_2 is the upper bound. This is reflective of the genetically ingrained habit and morphology of the individual species [22].

After executing the three update steps, the radius and height are incremented according to the computed ratio.

C. Inter-Plant Dynamics

In [4], plants were treated independently of one another and relationships between different plant types were not accounted for. To address this, we add companionship relationships between plant types which dictate growth patterns of individual plants dependent on their placement relative to others.

Both above and below ground interactions influence the companionship relationship factor between two plant types, which can be positive or negative [23]–[25]. Example of above ground interactions include changes to the physical environment such as providing shade, protecting against weather damage, and supplying structural support. Below ground interactions include providing nitrogen which fertilizes the soil, root-root activity and allelopathy, which occurs when a plant releases toxic chemicals that inhibit growth of other plants [5].

In AlphaGardenSim, we use the model described in [5] to account for companionship relations. Plant interrelationships are defined within the relationship matrix $\mathbf{C} \in \mathbb{R}^{k\times k}$, where k is the number of plant types in the garden. Here, $\mathbf{C}_{i,j}$ stores a value that describes the companionship between plants of type i and j.

The $\mathbf{C}$ matrix was populated by analyzing the growth curves of individual plants in the physical test bed relative to neighboring plants. One-hundred and twenty growth curves were created by annotating daily images of the garden with a plant's center and outermost radius. By comparing a plant's growth curve to the average growth curve of its type, we can discover if neighboring plants promote or hinder growth. Positive and negative scalar values were assigned and then tuned to minimize the MAE between simulated and real world individual plants.

The relationship matrix $\mathbf{C}$ is then used to calculate the companionship factor c. For a given plant i,

$$c_i = \sum_{j\in[1,\cdots,N], j\neq i} \frac{\mathbf{C}_{p(i),p(j)}}{\|l(i) - l(j)\|_2^2}$$

where $p(i)$ is the plant type of seed i and $l(i) = (x_i, y_i)$ as the location of seed i. The companionship factor is used to update the daily growth parameter, $\tilde{g}$, which is determined by water and light resource allocation. The new daily radial growth parameter is defined to be $g = \tilde{g} \cdot c$.

D. Plant Life Cycle

The plant life cycle consists of five non-overlapping stages: germination, vegetative, reproductive, senescence and death [27], [28]. The number of timesteps between consecutive stages is a random variable sampled from a plant-specific discritized Gaussian distribution, assuming that plants of the same type share transition times between stages [29].

1) Germination: Germination starts when the seed is planted. In this stage the plant occupies a single point in the garden and has 0 radius and height. It allocates resources according to the model described in Section IV-B, however it does not grow, maintaining 0 radius and height until it transitions to the next stage. The initial non-zero radius and height are random variables sampled from a plant-specific Gaussian distribution.

2) Vegetative: During the vegetative stage, the plant allocates resources and grows according to the model specified in Section IV-B, unless it experiences stress from over or underwatering, as further detailed in Section IV-E.

3) Reproductive: During the reproductive stage, the plant behaves similarly to the vegetative stage, except it does not change in radius or height, unless it experiences stress from over or underwatering, as further detailed in Section IV-E.

4) Senescence: During the senescence stage, the plant does not change in height, however it allocates less water than before and its radius decays exponentially as it is wilting. If w_d is the plant's desired water amount, throughout the senescence stage it is multiplied by a coefficient, so that the adjusted desired water amount $\widetilde{w}_d$ decreases linearly to 0 over time:

$$\widetilde{w}_d = \frac{1-t}{t_s} w_d$$

where t is the amount of time the plant has spent in the senescence stage, and t_s is the total duration of the senescence stage.

5) Death: When the plant dies, it stops allocating resources and does not change in radius or height. However, it continues to occupy space in the garden, potentially occluding plants.

E. Water Stress

AlphaGardenSim models the response of plants to sub-optimal irrigation, namely over and underwatering, during the two life stages in which the plant accumulates biomass: the vegetative and the reproductive stages. A plant receives sufficient irrigation if the following conditions are met:

$$w(t) \geq T_o \cdot w_d$$
$$\tilde{w}(t) \leq T_u \cdot w_d$$

where $w(t)$ and $\tilde{w}(t)$ are the total amount of soil moisture within the plant's radius and its water uptake, respectively, T_o and T_u are over and underwatering plant-specific threshold parameters, and w_d is the plant's desired water amount.

Otherwise, the plant enters into water stress, and its radius decays exponentially until it reaches a fraction of its radius and transitions to the death stage or it receives sufficient irrigation. In addition, the effects of water stress are visualized via the plant's color, becoming progressively more brown as it continues to be stressed.

F. Irrigation

AlphaGardenSim uses a discrete-time linear approximation of Richards equation proposed by Tseng *et al.* [30] to model irrigation actions and soil moisture dynamics. As described in [4], the soil moisture model is defined as follows:

$$w(x, y, t) = \max(w(x, y, t-1) - f + a_w(x, y, t) - u(x, y, t), 0)$$

AlphaGardenSim uses the previous soil moisture content $w(x, y, t-1)$, the amount of irrigation applied $a_w(x, y, t)$, plant water uptake $u(x, y, t)$, and local water loss f to calculate the current soil moisture value for each discrete grid point $p(x, y)$ at time t.

To more accurately model water dynamics in AlphaGardenSim, we conducted physical test bed experiments using six TEROS-10 [31] volumetric water content soil moisture sensors connected to a ZL6 Data Logger [31]. These experiments were used to refine the parameters $a_w(x, y, t)$, $w(x, y, t-1)$, and f in the soil moisture model. We intend to tune plant uptake, $u(x, y, t)$, in future real world experiments.

We first made modifications to the irrigation application parameter, $a_w(x, y, t)$, by carrying out experiments with the FarmBot watering nozzle. Using a compartmentalized container placed beneath the nozzle, we discovered the area of influence of a watering action to be concentrated within a circle of 0.04m radius. Furthermore, we identified the FarmBot nozzle to have a flow rate of 0.083 L/s. In AlphaGardenSim, $a_w(x, y, t)$ is set to 0.200 L.

Through the use of the TEROS-10 moisture sensors, we were then able to determine a model for radial flow, or spread, of water once in the soil. To discover this radial flow model, we conducted a set of experiments in which the FarmBot watered at incremental distances from the center of a soil moisture sensor, beginning directly overhead, and ending at 0.10m away. Once outside of the 0.04m radius in which water is applied, the moisture gain is roughly halved at each subsequent 0.01m when compared to the water gain within the radius. Beyond 0.09m, we found no substantial gain. Thus, we found $\Delta w(x_r, y_r) = (1/2)^r * gain$ where r is distance measured in 0.01m outside of the 0.04m radius, (x_r, y_r) is a point $r + 0.04$m away from (x, y), $w(x, y)$ is the soil moisture at point (x, y), and $gain$ is the moisture gain for soil directly under the nozzle.

Next, we used the moisture sensors to tune the local water loss parameter, f, and build upon our findings in [5]. By watering at varying frequencies over the TEROS-10 sensors and with a set volume, we were able to plot water loss over time curves. One such water loss curve can be seen in Fig. 4. As the simulator operates on a day to day time-scale, the water loss we care to discover is that over one or more days after watering. In an experiment conducted in the physical garden bed, we directly watered 0.200L over five independent sensors at the same time every day. We model water loss and gain over each square grid point, with side length of 0.01m, in AlphaGardenSim by sampling from a univariate Gaussian calculated from experimental data.

We calculated the daily loss by averaging the loss of all five sensors. For each sensor, the daily loss was the difference between the highest value recorded by the sensor within three hours after watering, in which soil moisture gain occurred, and the lowest value recorded directly before the next watering action. The Gaussian for water loss over a single day has a mean of 0.042 m^3/m^3 and a standard deviation of 0.0048 m^3/m^3, and f is sampled from such in AlphaGardenSim. For more than one day after watering, the Gaussian for loss has a mean of 0.01 m^3/m^3 and a standard deviation of 0.0001 m^3/m^3. Similarly, we modeled the $gain$ as a univariate Gaussian calculated from daily gains over a two week span, where the daily gain was calculated by taking the difference between the value right before and within three hours after watering on the same day. The Gaussian for gain has a mean of 0.046 m^3/m^3 and a standard deviation of 0.0054 m^3/m^3 and was set accordingly in simulation.

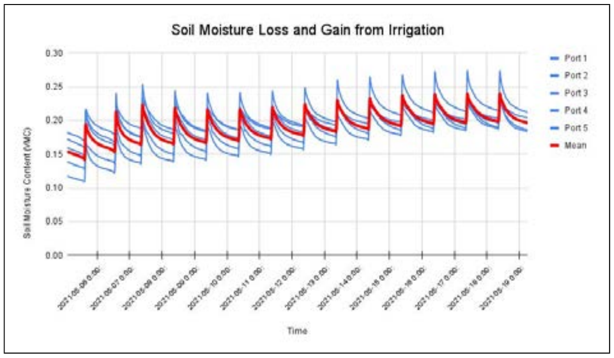


Fig. 4. Soil moisture curve generated from TEROS-10 soil moisture sensors connected to a data logger to determine water loss and gain rates. Irrigation was applied every 24 hours. Soil moisture readings were recorded every 30 minutes. The five blue curves represent five different sensors that were each watered independently. The red curve is the average of the readings of all five sensors.

Moreover, we tuned the prior soil moisture content parameter, $w(x, y, t-1)$, using another set of experiments with the TEROS-10 sensors. To do so, we identified the maximal volumetric water content of our specific soil, which indicates the water storage capacity of the medium. In the experiments, we saturated five different samples of soil using varying watering techniques and discovered the max volumetric water content to be around 0.3. Thus, we capped both $w(x, y, t-1)$ and $w(x, y, t)$ at this value.

Through the execution of physical test bed experiments and the utilization of soil moisture sensors and the FarmBot watering nozzle, we were able to tune parameters of the AlphaGardenSim irrigation model to more realistically simulate the characteristics of the real world garden.

V. Pruning, Irrigation, and Planting Policies

We evaluate the performance of different polyculture pruning, irrigation and planting policies by assessing their robustness in varying garden settings to achieve high plant yield and reduce water use in AlphaGardenSim.

A. Policies

We implement five policies:

1) Uniform Policy, a policy that irrigates according to a fixed schedule and prunes all plants uniformly.
2) Fixed Pruning, a policy that irrigates and prunes plants with a fixed pruning level based on water availability, plant health and garden diversity.
3) Variable Pruning, a policy that selects a pruning level $p \in \mathcal{P}$ for each day t from a discrete set of pruning levels $\mathcal{P}$.
4) Learned Pruning, a deep supervised learned policy that learns from Variable Pruning prune level demonstrations to predict prune levels over 1500X faster than Variable Pruning.
5) Dynamic Planting, a policy that seeds plants throughout the lifespan of the garden to achieve indefinite garden growth.

1) Uniform Policy: Introduced in [4], Uniform Policy irrigates all plants every other day similar to an array of drippers or sprinklers in farms and greenhouses. To limit overcrowding, every 5 days, the policy prunes all plants that grew beyond a threshold with $p = 5\%$.

2) Fixed Pruning: In [4], we presented Fixed Pruning, which utilizes soil moisture, plant health and global diversity to dynamically prune and irrigate each sector it observes. For every $\mathbf{o}(x, y, t)$, Fixed Pruning applies one of four actions: irrigate, prune, irrigate and prune, or none.

If any of the plant health values $h(x, y, t)$ in $\mathbf{o}(x, y, t)$ within the radial distribution of the water nozzle described in IV-F indicates underwatered, the policy irrigates the sector. If the sector does not contain any plants or only dead plants, Fixed Pruning does not irrigate. To avoid irrigating plants that are overwatered, the policy sums all $w(x, y, t)$ in the sector and doubles $w(x, y, t)$ wherever $h(x, y, t)$ contains an overwatered plant. If the total sum is less than a threshold, the sector is irrigated.

Fixed Pruning selects a pruning action if the proportion of any plant type, calculated by $r_d(t)$, in the pruning window is greater than a uniform threshold.

3) Variable Pruning: Experiments in prior work [5] and those in Section VI-D2 suggest that, due to a fixed pruning level, Fixed Pruning struggles to manage plants with significant differences in germination times, maturation times and max radii. To address this limitation, we introduced Variable Pruning, a policy that selects a pruning level $p \in \mathcal{P}$ for each day t from a discrete set of six pruning levels $\mathcal{P}$. Every timestep, Variable Pruning takes a 1-step lookahead to simulate the potential multi-modal entropy mme (see VI-B) that would result from choosing pruning level $p_i \in \mathcal{P}$ on the current garden state. After selecting p, Variable Pruning uses Fixed Pruning to collect pruning and irrigation actions for every $\mathbf{o}(x, y, t)$.

4) Learned Pruning: We introduced Learned Pruning in [5], as a way to speed-up 1-step lookahead with Variable Pruning by over 1500X. We train a deep supervised learned policy, mapping prune level p demonstrations from Variable Pruning to full garden states as illustrated in Fig. 5. A deep CNN with 18,244 parameters takes in an RGB garden overhead observation, a matrix of plant health, plant types, and water availability, and the global population distribution to determine a prune level for a plant.

5) Dynamic Planting: Dynamic Planting is an extension of Variable Pruning that uses the new planting action defined in Section III to obtain continuous coverage over longer garden periods, past the days of when plants seeded on day 0 live. We wish to seed plants in locations that minimize inter-plant competition for light and water so we provide the policy vacancy scores $e(x, y, t)$ for all (x, y) in each $\mathbf{o}(x, y, t)$. If any $e(x, y, t)$ in $\mathbf{o}(x, y, t)$ is above a threshold, and the maximum number of plants the policy can seed each day has not been reached, the policy seeds a plant at that location.

Dynamic Planting has several benefits over other policies that use stagnant seed placements. Dynamic Planting has potential to limit plant competition and achieve higher diversity due to the fact that smaller, slow growing plants can be seeded prior to larger, fast growing plants when the garden period begins. Furthermore, a garden period is no

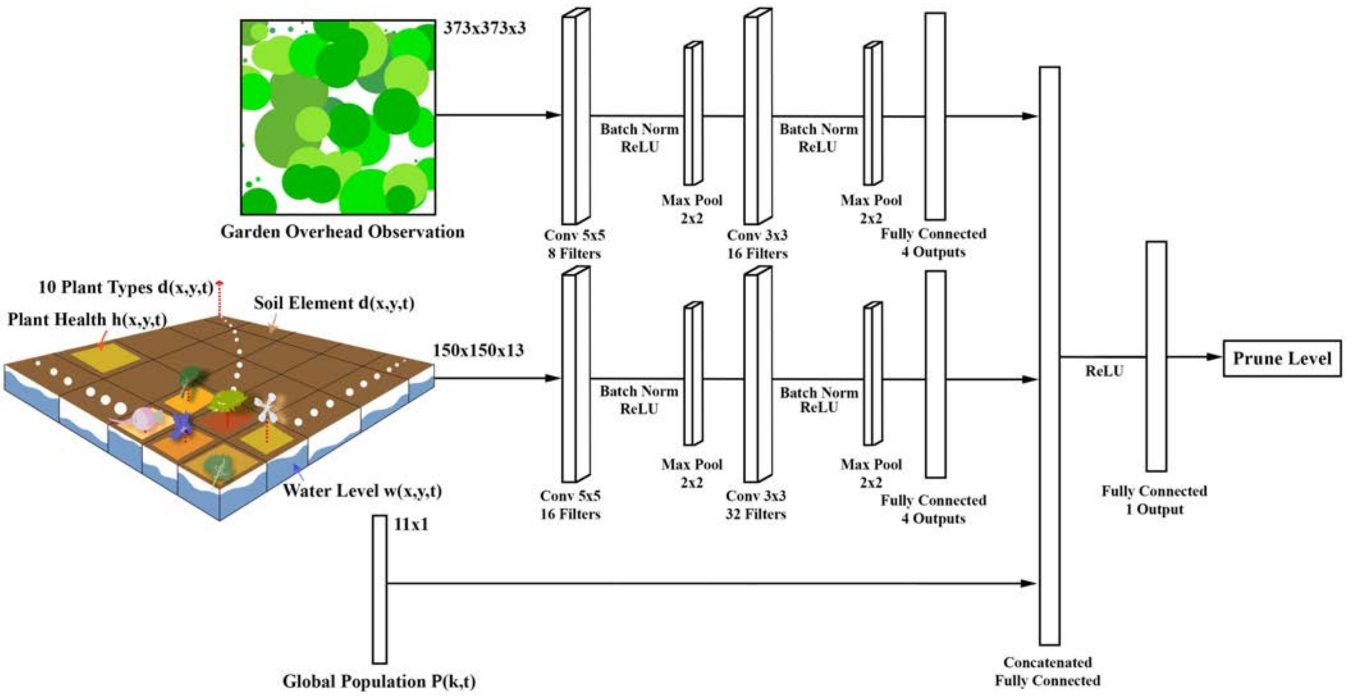
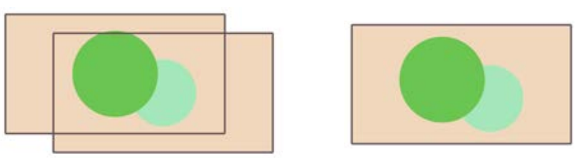


Fig. 5. Learned Pruning network architecture. A deep convolutional neural network with 18,244 parameters. The network takes three inputs: 1) an RGB image of the full garden; 2) a matrix of $h(x, y, t)$, $w(x, y, t)$ and $\mathbf{d}(x, y, t)$ for all (x, y) in the garden; 3) the global population distribution $\mathbf{P}(k, t)$ including soil coverage. The network predicts a prune level for each observation using demonstrations from Variable Pruning.



(a) Overlapping Sectors (b) Adaptive Sector Sampling

Fig. 6. **Adaptive Sector Sampling.** In [4], we introduced a sector observation method which sampled sectors centered around each seed location $s(x, y)$. However, this can lead to overlap in which multiple irrigation and pruning actions are taken when instead only one is needed. **Left:** Two plants are close to each other, so sectors centered around each of their $s(x, y)$ overlap. **Right:** With adaptive sector sampling, if the plants are both germinating or are both growing, their seed locations are clustered and a sector centered at the center of the cluster is instead observed.

longer constrained by constant companionship relations; new plants that are seeded can be chosen through a combination of optimizing local companionship relations and to improve global diversity and coverage.

B. Adaptive Sector Sampling

In prior work [4], we introduced a sector sampling method which, at every timestep, samples m sectors centered at each $s(x, y)$ and an additional $\frac{m}{10}$ sectors centered at non-seed points. However, as seen in Fig. 6, sectors can overlap due to plants seeded close to each other. During irrigation, both sectors may be watered, resulting in extra water usage. Additionally, multiple pruning actions may be used instead of one to prune all plants in the overlapping area. To address this, we create clusters of seed locations $s(x, y)$ that are within a distance c_d of each other. We center observations at the centers of these clusters to encompass all plants within that cluster. We create two sets of clusters: the seed locations of germinating plants that are within $c_{d,germ}$ of each other, and the seed locations of growing plants that are within $c_{d,grow}$ of each other. To further reduce the number of actions, we do not cluster, and consequently do not irrigate or prune, the seed locations of plants in Senescence or Death as these two stages are irreversible.

VI. Experiments

A. Experimental Setup

We conduct experiments on 150×150cm gardens with 100 plants sampled with replacement from k plant types. Plants are seeded at random locations $s(x, y) = \mathbf{d}(x, y, 0)$. To promote plant germination and early growth, we set the overwatering threshold to $T_o = 100$ as it represents the maximum amount of water in a 10×10cm square around a plant. The underwatering threshold is set to $T_u = 0.1$.

B. Evaluation

We evaluate each policy on randomly seeded experiments, using the following metrics:

1) Average Total Plant Coverage - We average the total percent coverage in a single experiment over days 20 to 70 of the growing period, taking into account only the coverage of the plants, ignoring the uncovered space labeled as *soil*:

$$AC = \frac{\sum_t r_c(t)}{T}$$

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

2) Average Diversity - We average the diversity in a single experiment between days 20 and 70 of the growing period, $T = 50$:

$$AD = \frac{\sum_t r_d(t)}{T}$$

During the beginning and end of the growing period, the diversity is always high, since all plants are very small (germinating or dying). Therefore, these diversity measurements are not reflective of the policy's performance.

3) Water Usage (liters) - We sum the water used in a single experiment over the entire growing period (100 days):

$$WU = \sum_t -r_w(t)$$

4) Multi-Modal Entropy - We model diversity as the normalized entropy of $\mathbf{P}(k, t)$ for all k plant types. Maximum diversity, hence, equates to a uniform distribution within $\mathbf{P}(k, t)$. However, prior work [5] and Section VI-D2 suggests that high diversity can be achieved with low plant coverage and consequently, high soil exposure. Thus, we define $\tilde{k}$ be the union of the k plant types and an additional type representing the amount of unoccluded soil, so that $\mathbf{P}(\tilde{k}, t)$ will include soil coverage. We define multi-modal entropy (mme) as:

$$mme(t) = \frac{H\big(\mathbf{P}(\tilde{k}, t)\big)}{\log \tilde{k}} = \frac{-\sum_{i=1}^{\tilde{k}} \mathbf{P}(i, t) \log \mathbf{P}(i, t)}{\log \tilde{k}}$$

C. Adaptive Sector Sampling Experiments

We compare the evaluation metrics achieved with adaptive sector sampling versus the sector observation approach from [4]. Gardens contain 100 plants, 10 of each plant type from Table I. As described in Section IV, the amount of water a plant receives while growing after germination corresponds to how much it grows. A smaller $c_{d,grow}$ results in more sectors observed for growing plants as less plants are within the threshold for clustering. A $c_{d,grow} = 1$cm resembles the sector sampling approach from [4]. We experiment with two $c_{d,grow}$ clustering thresholds: 2cm and 8cm. We fix $c_{d,germ}$ to 8cm as germination duration does not depend on the amount of water provided. Results averaged across 20 gardens over 100 days are summarized in Table III. With a $c_{d,grow} = 2$cm, Fixed Pruning observes more sectors and provides more water to each plant than $c_{d,grow} = 8cm$. However, both thresholds achieve comparable coverage, diversity and mme to the sector observation approach from [4]. By clustering germinating and growing plants that are $c_{d,germ} = 8$cm and $c_{d,grow} = 2$cm of each other, Fixed Pruning uses 37% less water over the entire garden simulation period, 38% less irrigation actions, and 35% less pruning actions. With a $c_{d,grow} = 8$cm threshold, Fixed Pruning is able to use 50% less water, irrigation actions and pruning actions.

D. Pruning and Irrigation Experiments

1) Table I Plant Types: We evaluate Uniform Policy, Fixed Pruning, and Variable Pruning on gardens with 100 edible plants, 10 plants from each of the 10 plant types in Table I. Observations to policies are determined through adaptive sector sampling described in Section V-B with $c_{d,germ} = 8$cm and $c_{d,grow} = 8$cm.

Through experiments, we found that a fixed prune level of 15% for Fixed Pruning and the set of pruning levels $\mathcal{P} \in (5\%, 10\%, 16\%, 20\%, 30\%, 40\%)$ for Variable Pruning, leads to high coverage, diversity and mme on the plant set. Results averaged across 20 different random garden seed placements over 100 garden days are summarized in Table IV. Compared

TABLE III

POLICY EVALUATIONS OF FIXED PRUNING AVERAGED ACROSS 20 TEST GARDENS DURING DAYS 20 TO 70 WITH AND WITHOUT ADAPTIVE SECTOR SAMPLING. WE OBSERVE GERMINATING PLANTS WITHIN 8cm OF EACH OTHER IN THE SAME OBSERVATION SECTOR AND EXPERIMENT WITH 2cm AND 8cm CLUSTER RADII FOR GROWING PLANTS. BOTH A 2cm AND 8cm CLUSTER RADII FOR GROWING PLANTS ARE ABLE TO ACHIEVE COMPARABLE COVERAGE, DIVERSITY AND MULTI-MODAL ENTROPY TO THE SECTOR OBSERVATION APPROACH FROM [4]. WHILE $c_{d,grow} = 2$cm USES OVER 35% LESS WATER, IRRIGATION ACTIONS AND PRUNING ACTIONS THAN WITHOUT ADAPTIVE SECTORING, $c_{d,grow} = 8$cm USES OVER 50% LESS WATER AND ACTIONS

Metric	Without	$c_{d,grow} = 2$cm	$c_{d,grow} = 8$cm
Avg coverage	0.71	**0.74**	0.71
Avg diversity	0.91	0.91	0.91
Avg multi-modal entropy	0.83	**0.84**	0.82
Avg water use (liters)	15.73	9.80	**7.36**
Num. of irrigation actions	7862.6	4900.5	**3680.9**
Num. of pruning actions	732.3	474.0	**292.5**

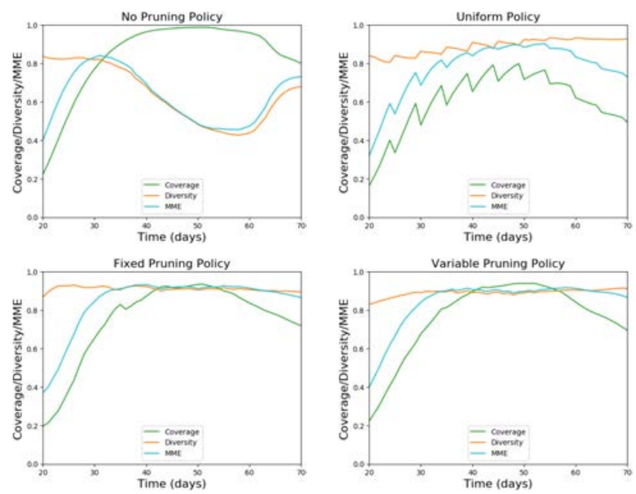


Fig. 7. **Simulation Experiments with Table I Plant Types.** The total plant coverage and diversity achieved by Uniform Policy, Fixed Pruning, and Variable Pruning and a no pruning baseline. Figures are from a garden during days 20 and 70 of a 100 day simulation. Metrics are calculated between these days as policies do not prune until day 20 and plants begin to die after day 70. Observations to policies are provided with adaptive sector sampling from V-B. The no pruning policy is able to achieve high coverage at the expense of low diversity. Uniform Policy manages to increase the diversity, but overprunes plants every 5 days due to not being able to prune every day. Fixed Pruning and Variable Pruning, by being able to prune and irrigate plants every day, both achieve high coverage, diversity, and consequently, higher multi-modal entropy than Uniform Policy.

TABLE IV

POLICY EVALUATIONS OF UNIFORM POLICY, FIXED PRUNING, VARIABLE PRUNING, AND LEARNED PRUNING AVERAGED ACROSS 20 TEST GARDENS EACH WITH 100 PLANTS. **TOP 4 ROWS:** EXPERIMENTS USE THE 10 PLANT TYPES FROM TABLE I. METRICS ARE AVERAGED BETWEEN DAYS 20 TO 70 AS POLICIES DO NOT PRUNE PRIOR TO DAY 20 AND PLANTS BEGIN TO DIE AFTER DAY 70. **BOTTOM 5 ROWS:** EXPERIMENTS USE THE 10 PLANT TYPES FROM TABLE V. THE FASTER GROWING PLANTS BEGIN TO DIE AFTER DAY 50, SO WE INSTEAD AVERAGE METRICS FOR THESE GARDENS BETWEEN DAYS 20 TO 50. THE COMPUTATION TIME REPRESENTS THE TIME IT TAKES A POLICY TO COMPUTE AN ACTION GIVEN AN OBSERVATION. THE VARIABLE PRUNING IS COMPUTATIONAL INTENSIVE AS IT EVALUATES DIFFERENT PRUNING LEVELS, WHILE LEARNED PRUNING PERFORMS SIMILARLY BUT HAS A SIGNIFICANTLY LOWER COMPUTATION TIME

Metric	Uniform	Fixed	Variable	Learned
Avg coverage	0.59	0.71	**0.74**	-
Avg diversity	0.88	**0.91**	0.88	-
Avg multi-modal entropy	0.75	0.82	0.82	-
Avg water use (liters)	7.53	7.36	**7.34**	-
Avg coverage	-	0.23	**0.46**	0.42
Avg diversity	-	**0.76**	0.67	0.67
Avg multi-modal entropy	-	0.37	**0.55**	0.53
Avg water use (liters)	-	**18.67**	19.81	19.80
Computation time (seconds)	-	-	336.18	**0.22**

to a baseline no pruning policy which irrigates every other day, Uniform Policy achieves higher diversity but overprunes plants due to only being able to prune every 5 days. Both Variable Pruning and Fixed Pruning achieve higher coverage, diversity, multi-modal entropy than Uniform Policy by dynamically selecting which plants to irrigate and prune each day based on plant health, garden diversity, and multi-modal entropy for Variable Pruning.

2) Table V Plant Types: As suggested in prior work [4], [5], Fixed Pruning struggles to achieve both high coverage and diversity, and consequently high mme on plants with different germination times, maturation times and max radii. To illustrate this, we conduct experiments on gardens with 100 plants, 10 plants from each of the 10 plant types in Table V where faster growing plants grow 5X to 8X faster than slower ones. We use the observation sampling method described in prior work [4], sampling m sectors centered at each $s(x, y)$ and $\frac{m}{10}$ sectors centered at non-seed points.

We simulate Fixed Pruning with 15% and 1% pruning levels, and Variable Pruning with prune levels $\mathcal{P} \in (5\%, 10\%, 16\%, 20\%, 30\%, 40\%)$. To achieve higher plant diversity, Fixed Pruning with a 15% prune level aggressively prunes the faster growing plants during their growing period to match the size of the slower growing plants. The policy achieves high diversity at the cost of low coverage, resulting in low multi-modal entropy. Fixed Pruning with a 1% prune level prunes the faster plants less, but fails to prune enough to achieve uniform plant diversity. As a result a 1% prune level also results in low multi-modal entropy.

Comparing Fixed Pruning with a 15% prune level against Variable Pruning, we simulate both policies on 20 randomly

TABLE V

FAST AND SLOW PLANT TYPES. AVERAGE GERMINATION TIME, MATURATION TIME, AND MAX RADII OF 5 FAST AND 5 SLOW GROWING PLANT TYPES. WE EXPERIMENTED WITH VARYING GERMINATION TIMES, MATURATION TIMES AND MAX RADII TO CREATE THE PLANT SET ABOVE WHERE UNIFORM POLICY ACHIEVES LOW MULTI-MODAL ENTROPY mme, AS ILLUSTRATED IN FIG. 8

Plant Type	Germination (days)	Maturation (days)	Max Radius (cm)
Fast growing	9.8	20.2	93.3
Slow growing	25.6	90.4	31.95

seeded gardens. Results are averaged in Table IV. The faster growing plants from Table V begin to die after day 50 so results are averaged between days 20 and 50. Variable Pruning, selecting prune levels that would result in the highest mme 1-day into the future, initially uses a small prune level of 5% to allow faster growing plants to grow. As the faster growing plants begin to die, Variable Pruning uses higher prune levels to increase diversity and maintain high multi-modal entropy. As a result, Variable Pruning achieves high coverage, diversity and multi-modal entropy.

Due to the long runtime of simulating 1-step lookahead with Variable Pruning, we train Learned Pruning to learn prune levels for each day given full garden states of gardens with slow and fast growing plants from Table V. We simulate Variable Pruning on 6,500 gardens with randomized seed locations to collect prune level demonstrations between days 20 to 100 as policies do not prune before day 20. Variable Pruning selects a 5% prune level 95% of the time and levels 10% and greater 5% of the time to maximize multi-modal entropy for the slow and fast plant types. We increase the number of demonstrations for prune levels 10% and greater by 8X by rotating and flipping observations. The network is trained with 520K demonstrations for 55 epochs with the Adadelta [32] optimizer and mean squared error loss using 4 hardware threads and 4 Tesla V100 GPUs. The network architecture and optimization framework is written in Python using PyTorch. Table IV summarizes results averaged across 20 test gardens withheld from the training dataset. Learned Pruning achieves comparable coverage, diversity, mme and water usgae to Variable Pruning but is over 1500X faster predicting p for days 20 to 50.

E. Dynamic Planting Experiments

We conduct Dynamic Planting experiments on a general setting initially consisting of 100 plants from 10 types. To evaluate how well dynamic planting can sustain garden growth, we simulate a growing period of 200 days. We follow the observation method from [4] to allow the policy to observe locations away from seed points $s(x, y)$. Dynamic Planting begins seeding plants after day 20, which is when most of the original plants have reached the vegetative stage. The policy uses a vacancy threshold of $e(x, y, t) = 8$cm and can seed a maximum of 5 plants every day. Results averaged across 10

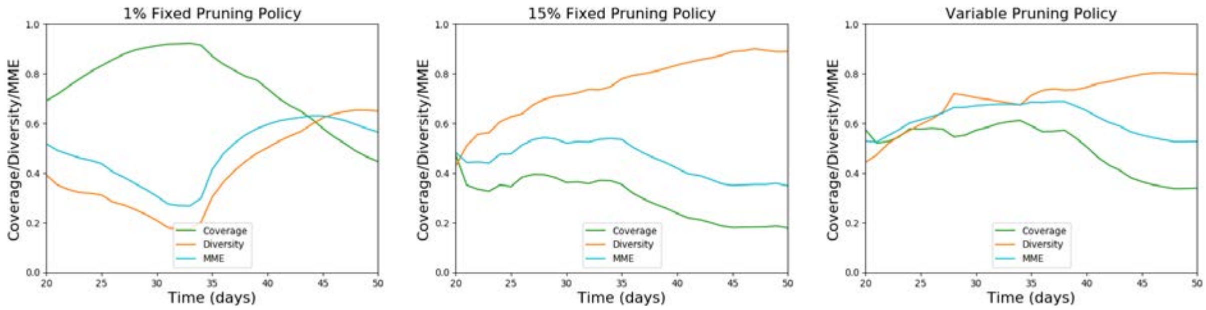


Fig. 8. Simulation results on gardens between days 20 and 50 with the fast and slow growing plant types from Table V. Metrics are shown between days 20 and 50 as the faster growing plant types begin to die after day 50. **Left:** Simulation results for Fixed Pruning with fixed prune levels of 1%. With a 1% fixed prune level, Fixed Pruning achieves high coverage but struggles to maintain diversity. As a result multi-modal entropy is low. **Middle:** With a 15% prune level, Fixed Pruning achieves high diversity but low coverage as a result of pruning the fast growing plants to match the size of the slower plants. **Right:** Variable Pruning simulation results. By optimizing for multi-modal entropy, the policy is able to manage both coverage and diversity through variable prune levels and achieve the highest multi-modal entropy. During earlier days, Variable Pruning uses smaller prune rates to allow the faster growing plants the grow. As the fast plants begin to die, to maintain high multi-modal entropy, Variable Pruning prunes more frequently.

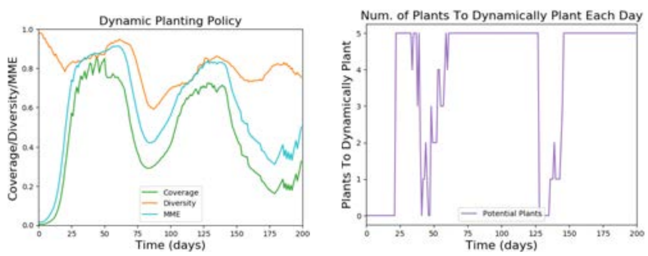


Fig. 9. **Dynamic Planting Policy.** The policy seeds up to 5 new plants every day after day 20. During periods where coverage is high in the garden, there is little vacant space to seed new plants. As a result, the number of plants selected to be dynamically planted drops during days 35 to 61 and days 128 to 146. After these high coverage periods, up to 5 new plants are seeded every day resulting in a resurgence in coverage after the new plants germinate and mature.

TABLE VI

DYNAMIC PLANTING POLICY AVERAGED ACROSS 10 TEST GARDENS WITH 100 INITIAL PLANTS AND THE ABILITY TO SEED UP TO 5 NEW PLANTS EVERY DAY AFTER DAY 20. EVALUATION METRICS ARE AVERAGED ACROSS ALL 200 DAYS OF GARDEN SIMULATION. RESULTS SHOW THAT REPLANTING SEEDS CAN LEAD TO SUSTAINED GROWTH AND DIVERSITY ACROSS INDEFINITE PERIODS OF TIME

Policy	Coverage	Diversity	MME	Water Use (liters)
Dynamic Planting	0.50	0.82	0.63	158.98

test gardens with random seed placements are summarized in Table VI and visualized in Fig. 9. Since Dynamic Planting only seeds new plants in locations that are sufficiently vacant, during periods where coverage is high in the garden, the number of plants seeded every day drops below 5. Once plants begin to die and coverage decreases, the garden becomes sparser, allowing Dynamic Planting to find locations where vacancy $e(x, y, t) \geq 8$cm. After the new plants germinate and mature, coverage rebounds.

VII. DISCUSSION AND FUTURE WORK

This paper is a revised and greatly expanded version of two prior conference papers [4], [5]. We present an irrigation model, tuned from real world measurements to account for soil moisture gain and loss. We extend the action space first presented in [4] by allowing the agent to plant seeds dynamically to increase the resources utilization and extend the garden lifetime. We also develop an adaptive sampling method to reduce the number of actions taken by the agent and increase its efficiency. We evaluate automation policies in simulation using a novel metric to maximize the garden's leaf coverage and plant diversity. In future work, we will further tune the models for plant water uptake and vertical growth with real world measurements, and evaluate the performance of the trained policies on real gardens.

ACKNOWLEDGMENT

This research was performed at the AUTOLAB at the UC Berkeley in affiliation with the Berkeley AI Research (BAIR) Laboratory, Berkeley Deep Drive (BDD), the Real-Time Intelligent Secure Execution (RISE) Laboratory, and the CITRIS "People and Robots" (CPAR) Initiative. The authors were supported in part by donations from Siemens, Google, Toyota Research Institute, Honda Research, and Hewlett-Packard, a fellowship within the IFI Program of the German Academic Exchange Service (DAAD).

The authors thank their colleagues who provided helpful feedback and suggestions, in particular Mary Power, Jackson Chui, Jeff Ichnowski, Micah Carroll, Paul Shao, Eric Siegel, Isaac Blankensmith, Maya Man, Sarah Newman, Shubha Jagannatha, Sona Dolasia, Christiane Paul, Vishal Satish, Raven Huang, and Atsunobu Kotani.

REFERENCES

[1] S. Gliessman *et al.*, "Polyculture cropping has advantages," *California Agricult.*, vol. 36, no. 7, pp. 14–16, 1982.

[2] M. Liebman, "Polyculture cropping systems," in *Agroecology*. Boca Raton, FL, USA: CRC Press, 2018, pp. 205–218.

[3] T. E. Crews, W. Carton, and L. Olsson, "Is the future of agriculture perennial? Imperatives and opportunities to reinvent agriculture by shifting from annual monocultures to perennial polycultures," *Global Sustainability*, vol. 1, pp. 1–18, Sep. 2018.

[4] Y. Avigal *et al.*, "Simulating polyculture farming to tune automation policies for plant diversity and precision irrigation," in *Proc. IEEE 16th Int. Conf. Automat. Sci. Eng. (CASE)*, Aug. 2020, pp. 238–245.

[5] Y. Avigal *et al.*, "Learning seed placements and automation policies for polyculture farming with companion plants," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2021, pp. 902–908.

[6] J. W. Jones *et al.*, "The DSSAT cropping system model," *Eur. J. Agron.*, vol. 18, nos. 3–4, pp. 235–265, 2003.

[7] P. Steduto, T. C. Hsiao, D. Raes, and E. Fereres, "AquaCrop—The FAO crop model to simulate yield response to water: I. Concepts and underlying principles," *Agron. J.*, vol. 101, no. 3, pp. 426–437, May 2009.

[8] K. R. Thorp and K. F. Bronson, "A model-independent open-source geospatial tool for managing point-based environmental model simulations at multiple spatial locations," *Environ. Model. Softw.*, vol. 50, pp. 25–36, Dec. 2013.

[9] N. Chebrolu, T. Läbe, and C. Stachniss, "Spatio-temporal non-rigid registration of 3D point clouds of plants," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Aug. 2020, pp. 3112–3118.

[10] C. Damgaard, J. Weiner, and H. Nagashima, "Modelling individual growth and competition in plant populations: Growth curves of chenopodium album at two densities," *J. Ecol.*, vol. 90, no. 4, pp. 666–671, Aug. 2002.

[11] W. J. Price, B. Shafii, and D. C. Thill, "An individual-plant growth simulation model for quantifying plant competition," in *Proc. Conf. Appl. Statist. Agricult.*, Apr. 1994, pp. 1–18.

[12] U. Berger, C. Piou, K. Schiffers, and V. Grimm, "Competition among plants: Concepts, individual-based modelling approaches, and a proposal for a future research strategy," *Perspect. Plant Ecol., Evol. Systemat.*, vol. 9, nos. 3–4, pp. 121–135, Mar. 2008.

[13] T. Czárán and S. Bartha, "The effect of spatial pattern on community dynamics; A comparison of simulated and field data," in *Progress in Theoretical Vegetation Science*. Dordrecht, The Netherlands: Springer, 1990, pp. 229–239.

[14] M. Wiggert *et al.*, "RAPID-MOLT: A meso-scale, open-source, low-cost testbed for robot assisted precision irrigation and delivery," in *Proc. IEEE 15th Int. Conf. Automat. Sci. Eng. (CASE)*, Aug. 2019, pp. 1489–1496.

[15] N. Habibie, A. M. Nugraha, A. Z. Anshori, M. A. Ma'sum, and W. Jatmiko, "Fruit mapping mobile robot on simulated agricultural area in Gazebo simulator using simultaneous localization and mapping (SLAM)," in *Proc. Int. Symp. Micro-NanoMechtron. Hum. Sci. (MHS)*, 2017, pp. 1–7.

[16] E. Rohmer, S. P. N. Singh, and M. Freese, "V-REP: A versatile and scalable robot simulation framework," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 1321–1326.

[17] K. R. Aravind and P. Raja, "Design and simulation of crop monitoring robot for green house," in *Proc. Int. Conf. Robot., Current Trends Future Challenges (RCTFC)*, 2016, pp. 1–6.

[18] C. T. de Wit, "Photosynthesis of leaf canopies," Pudoc, Wageningen, The Netherlands, Tech. Rep. 663, 1965.

[19] T. C. Hsiao, "Effects of drought and elevated CO_2 on plant water use efficiency and productivity," in *Proc. Interacting Stresses Plants Changing Climate*. Berlin, Germany: Springer, 1993, pp. 435–465.

[20] A. Keller, "Evapotranspiration and crop water productivity: Making sense of the yield-ET relationship," in *Impacts Global Climate Change*, 2005, pp. 1–11.

[21] B. Dellen, H. Scharr, and C. Torras, "Growth signatures of rosette plants from time-lapse video," *IEEE/ACM Trans. Comput. Biol. Bioinform.*, vol. 12, no. 6, pp. 1470–1478, Nov. 2015.

[22] M. Iino, C. Long, and X. Wang, "Auxin- and abscisic acid-dependent osmoregulation in protoplasts of phaseolus vulgaris pulvini," *Plant Cell Physiol.*, vol. 42, no. 11, pp. 1219–1227, Nov. 2001.

[23] K. Adamczewska-Sowińska and J. Sowiński, "Polyculture management: A crucial system for sustainable agriculture development," in *Soil Health Restoration and Management*. Singapore: Springer, 2020, pp. 279–319.

[24] J. Parker, W. Snyder, G. Hamilton, and C. Rodriguez-Saona, "Companion planting and insect pest control," in *Weed Pest Control: Conventional New Challenges*. Rijeka, Croatia: InTech, Jan. 2013, pp. 1–30.

[25] A. L. Iverson *et al.*, "REVIEW: Do polycultures promote win-wins or trade-offs in agricultural ecosystem services? A meta-analysis," *J. Appl. Ecol.*, vol. 51, no. 6, pp. 1593–1602, Dec. 2014, doi: 10.1111/1365-2664.12334.

[26] P. G. Seeds. (2020). *Pinetree Garden Seeds—Vegetable Collections*. [Online]. Available: https://www.superseeds.com/

[27] A. Zardilis, A. Hume, and A. J. Millar, "A multi-model framework for the arabidopsis life cycle," *J. Exp. Botany*, vol. 70, no. 9, pp. 2463–2477, Apr. 2019.

[28] T. Hirose, T. Kinugasa, and Y. Shitaka, "Time of flowering, costs of reproduction, and reproductive output in annuals," in *Reproductive Allocation Plants*. Amsterdam, The Netherlands: Elsevier, 2005, pp. 159–188.

[29] P. J. Lumsden and A. J. Millar, *Biological Rhythms and Photoperiodism in Plants*. Oxford, U.K.: BIOS Scientific Publishers, 1998.

[30] D. Tseng *et al.*, "Towards automating precision irrigation: Deep learning to infer local soil moisture conditions from synthetic aerial agricultural images," in *Proc. IEEE 14th Int. Conf. Automat. Sci. Eng. (CASE)*, Aug. 2018, pp. 284–291.

[31] *METER Environment*. [Online]. Available: https://www.metergroup.com/environment/

[32] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," 2012, *arXiv:1212.5701*.

Yahav Avigal (Graduate Student Member, IEEE) received the B.Sc. degree from Tel-Aviv University. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering and Computer Science, UC Berkeley. He worked as a Software Engineer at Intel RealSense. He has been a Researcher with the UC Berkeley's AUTOLab since 2019. His current research interests include robot motion planning, object manipulation, and precision agriculture. He received the "Best Student Paper Award" in CASE 2020 for the paper "Simulating Polyculture Farming to Tune Automation Policies for Plant Diversity and Precision Irrigation."

William Wong received the bachelor's degree in computer science from the UC Berkeley, in May 2021. He is currently pursuing the M.S. degree in machine learning with Carnegie Mellon University. He joined the AUTOLab in 2018. He was the author on "Simulating Polyculture Farming to Tune Automation Policies for Plant Diversity and Precision Irrigation" and "Learning Seed Placements and Automation Policies for Polyculture Farming with Companion Plants," the former which won Best Student Paper Award at IEEE CASE 2020. His research interests include deep learning and AI for social good.

Mark Presten is currently pursuing the master's degree in electrical engineering and computer science with the UC Berkeley. He joined the AUTOLab in 2019. He has been researching the AlphaGarden project since 2019. He was the author of the paper titled, "Simulating Polyculture Farming to Tune Automation Policies for Plant Diversity and Precision Irrigation" that won the Best Student Paper Award at IEEE CASE 2020. His research interests include robotics, machine learning, agriculture, control, and design.

Mark Theis is currently an Undergraduate Researcher at the UC Berkeley, double majoring in EECS and mechanical engineering. Outside of academia, he worked in the semiconductor and space industries in both mechanical and electrical design roles. In 2020, he received the IEEE CASE Best Student Paper Award for "Simulating Polyculture Farming to Tune Automation Policies for Plant Diversity and Precision Irrigation." His research interests include robotics, networking, electronics and power systems, and hardware design.

Shrey Aeron (Student Member, IEEE) is currently pursuing the bachelor's degree in electrical engineering and computer science with the UC Berkeley. His research interests include robotics, natural language processing, and electronics design. Outside of research, he is interested in entrepreneurship, data science, and efficient transportation. He is an author of the paper titled "Trends in COVID-19 Publications: Streamlining Research Using NLP and LDA."

Anna Deza received the bachelor's degree from the University of Toronto. She is currently pursuing the Ph.D. degree with the Department of Industrial Engineering and Operations Research, UC Berkeley. Her thesis "A Multistage Stochastic Integer Programming Approach to Distributed Operation Room Scheduling" was selected as a Finalist for the INFORMS 2020 Undergraduate Operations Research Prize.

Satvik Sharma is currently pursuing the bachelor's degree in computer science with the UC Berkeley. He joined the AUTOLab, as an Undergraduate Researcher, in 2020. His research interests include robotics, machine learning, and reinforcement learning.

Rishi Parikh is currently pursuing the bachelor's degree in computer science with the UC Berkeley. He joined the AUTOLab in 2020. His research interests include robotics and artificial intelligence. Outside of academia, he is interested in augmented reality, and teaching.

Sebastian Oehme received the bachelor's degree in engineering science from TU Munich in March 2019, where he is currently pursuing the M.S. degree in electrical engineering and information technology. He joined Autolab in 2020 to collaborate on the AlphaGarden project. He was the author on "Affective Computing and Bandits: Capturing Context in Cold Start Situations" and "Learning Seed Placements and Automation Policies for Polyculture Farming with Companion Plants." His research interests broadly involve machine learning, automation, and affective computing.

Stefano Carpin (Senior Member, IEEE) received the "Laurea" and Ph.D. degrees in electrical engineering and computer science from the University of Padova, Italy, in 1999 and 2003, respectively. From 2003 to 2006, he held faculty positions with Jacobs University, Bremen, Germany. Since 2007, he has been with the School of Engineering, University of California at Merced (UC Merced), Merced, CA, USA, where he established and leads the robotics laboratory. He is currently a Professor with the Department of Computer Science and Engineering, UC Merced. His research interests include mobile and cooperative robotics for service tasks, and robot algorithms. He was the Founding Chair of the Department of Computer Science and Engineering, University of California at Merced. He served as an Associate Editor for the IEEE Transactions on Automation Science and Engineering and the IEEE Transactions on Robotics.

Joshua H. Viers is currently a Professor of Water Resources Management with the School of Engineering, UC Merced, where he also works as the Associate Dean for Research and the Campus Director of the Center for Information Technology Research in the Interest of Society (CITRIS). Since joining UC Merced, he has led campus efforts to build an experimental smart farm, to develop a joint industry-university consortium on ag-food-tech research, and to convene several universities in developing water accounting methods to secure a climate resilient water future. His research is multi-disciplinary by design, team science-based in execution, and dedicated to innovation in informatics and geospatial analysis in water and land management. He teaches environmental engineering with a focus on water resources management, geospatial analytics, and ecosystem restoration and sustainability.

Stavros Vougioukas (Senior Member, IEEE) received the Ph.D. degree from the Electrical, Computers and Systems Engineering Department, Rensselaer Polytechnic Institute, Troy, NY, USA, in 1995. After spending one year as a Post-Doctoral Researcher with the University of Parma, Italy, and five years working in industry, he joined the Agricultural Engineering Department, Aristotle University of Thessaloniki, Thessaloniki, Greece, as a Faculty Member, where he worked on agricultural robotics, from 2002 to 2011. He is currently a Professor with the Department of Biological and Agricultural Engineering, University of California at Davis, Davis, CA, USA, where he joined in 2012. He teaches and conducts research in the area of robotics and automation for agriculture. He has published 140 articles in refereed journals and conference proceedings, and holds two U.S. patents. He is a Fulbright Fellow.

Ken Goldberg (Fellow, IEEE) is currently a William S. Floyd Jr. Distinguished Chair in engineering at the UC Berkeley and a Chief Scientist at Ambidextrous Robotics. He is an artist, an inventor, and a roboticist. He co-founded the IEEE Transactions on Automation Science and Engineering. As the Director of UC Berkeley's AUTOLab, he trains the next generation of researchers and entrepreneurs. He and his students have published 300 peer-reviewed articles and nine U.S. patents. His artwork has been exhibited internationally and he founded the Art, Technology, and Culture Public Lecture Series in 1997. He is an Advisor to RoboGlobal ETF, and has presented over 500 invited lectures worldwide. More details can be found at: http://goldberg.berkeley.edu.