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

Yahav Avigal[1], Jensen Gao[1], William Wong[1], Kevin Li[1], Grady Pierroz[2,3], Fang Shuo Deng[1], Mark Theis[1], Mark Presten[1], and Ken Goldberg[1]

***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 than monoculture. As a first step toward developing automation control policies for polyculture farming, we present AlphaGardenSim, a fast, first order, open-access simulator that integrates single plant growth models with inter-plant dynamics, including light and water competition between plants in close proximity. The simulator approximates growth in a real greenhouse garden at 9,000X the speed of natural growth, allowing for policy parameter tuning. We present an analytic automation policy that in simulation reduced water use and achieved high coverage and plant diversity compared with other policies, even in the presence of invasive species. Code and supplementary material can be found at https://github.com/BerkeleyAutomation/AlphaGarden.**

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], [2], [3]. Polyculture is also more practical for confined urban spaces and essential for aesthetic gardens.

With increasing demands for fresh local herbs and produce, in spite of limited resources, polyculture gardens may be 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. Still, finding an optimal policy is a challenging task. The long time constants for real-world experiments motivates the use of a simulated environment, yet it is difficult to simulate inter-plant dynamics, including competition for light, water and nutrients. In addition, much of the interaction is done in the soil or beneath the leaf canopy and cannot be directly observed, and the action space, in particular pruning, is complicated.

[1]The AUTOLAB at UC Berkeley (automation.berkeley.edu) [2]Department of Plant and Microbial Biology, University of California, Berkeley, CA 94720 [3]Plant Gene Expression Center, US Department of Agriculture-Agricultural Research Service, Albany, CA 94710 {yahav_avigal, jenseng, williamwong, kevintli, gradypierroz, dfs.arthur, markt, mpresten, goldberg} @berkeley.edu

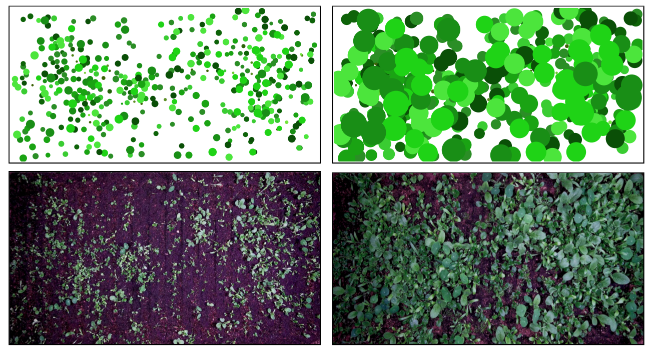


Fig. 1. **Simulated (top) and Physical (bottom) polyculture gardens.** Planted with seeds from 13 edible plant types, and grown for a period of 90 days. **Top:** A garden simulated with AlphaGardenSim, initialized with similar seed types and locations, and irrigated every 2 days for a similar period of time. Each shade of green in the simulated garden represents a different plant type. **Bottom:** A real garden of size 1.5x3 meters in a greenhouse with uniform light and irrigation. **Left:** The gardens at day 15, a few days after germination. **Right:** The gardens at day 30.

In this paper we present AlphaGardenSim, an efficient, first order simulator for polyculture farming. AlphaGardenSim models a polyculture garden using first order models to simulate single plant growth, inter-plant dynamics and competition for resources. This paper makes 3 contributions:

1) A fast, first order simulator, AlphaGardenSim, that incorporates parameterized individual plant growth models and inter-plant dynamics to simulate competition over resources between plants in close proximity.
2) An analytic automation policy that yields leaf coverage and garden diversity.
3) Application of the simulator to tune the policy parameters. Experimental results in simulation suggest that a tuned analytic policy can achieve high coverage and plant diversity compared with other policies, even in the presence of rapidly spreading invasive plants.

II. Related Work

Past work in plant growth simulation has predominantly been focused on monoculture agriculture. Widely used simulation models include DSSAT [4] and AquaCrop [5]. 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 [6] 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 plant level. Recently, Chebrolu et al. [7] 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. [8] proposed modeling competition between individual plants based on density and size differences between plants, 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. [9] 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. [10], 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 [11] 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, 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. [12] developed a testbed that enables real-time data collection of plant water stress to automate and optimize plant-level irrigation. Habibie et al. [13] 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 [14] 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 [15]. Even though each plant had unique parameters, they did not model inter-plant interactions.

III. AlphaGardenSim

In our simulator, AlphaGardenSim, the garden is modeled as a discrete $M \times N$ grid. The time t is specified in units of days. Each point $p(x, y)$ in the garden contains up to one seed, and up to a fixed capacity of water. We define $D(k)$ as a set of k plant types available in the garden, as well as types *soil* and *unknown*. At each point $p(x, y)$ in the garden at time t, we define $\mathbf{d}(x, y, t)$, a vector of length k representing the distribution of the plant (or soil) types that is visible overhead at point $p(x, y)$, i.e., the canopy cover in the garden. For example, when the garden is planted with plant type $D[3]$ at point $p(x, y)$ at time 0, then $\mathbf{d}(x, y, 0) = \left[0, 0, 1, 0, \ldots, 0\right]^T$. This can change over time if a plant from a neighboring point expands and occludes the plant seeded at $p(x, y)$. We define the seed locations in the garden as $\mathbf{s}(x, y)$, the health of the plant at point $p(x, y)$ at time t as $h(x, y, t)$, and the amount of soil moisture available at point $p(x, y)$ at time t as $w(x, y, t)$. We define the following global quantities over the entire garden; $\mathbf{p}(k, t)$ holds the global population in the garden as a distribution over point types D, $c(t)$ is the total canopy coverage w.r.t. the garden size at time t, $v(t)$ is the diversity in the garden at time t, and $s(t)$ represents the sustainability in the garden, i.e. the savings in irrigation compared to a uniform baseline irrigation policy.

IV. Modeling

A. Plant Growth Model

To represent each plant, we adopt the model proposed by [9] which tracks the radii of plants for the purposes of water competition. However, we add height as a simulated attribute, as this is necessary to model competition for sunlight. This yields efficient calculation of plant growth, while still modeling a variety of plant types and interactions. The overall growth curve of a plant is modeled as exponential in its radius, but with adjusted rates at each step when competition and limited resources are taken into account. This is consistent with experimental evidence suggesting that plants grow exponentially in ideal conditions [16].

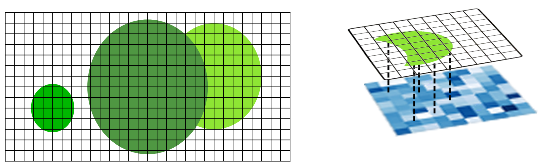
AlphaGardenSim executes a sequence of updates at each timestep: irrigation, lighting, water use and plant growth.

In the irrigation step, each of the $M \times N$ coordinate points accepts a water amount of 0 or 1. Values should be chosen in a garden-specific manner, such that 0 represents no water and 1 represents the maximum amount of water a unit area of soil can hold.

We assume a fixed amount of light is available at every point in the garden, so that plants receive light based on the size of their unoccluded leaf area (the portion of their bounding circle that is not covered by taller plants), as demonstrated in Fig. 2. To estimate the area of partially overlapping circles, we use an approximation. For a plant i with radius r_i, let L_i be the set of garden coordinates which

Local Variables				Global Variables			
$\mathbf{d}(x,y,t)$	$h(x,y,t)$	$w(x,y,t)$	$\mathbf{s}(x,y)$	$\mathbf{p}(k,t)$	$c(t)$	$v(t)$	$s(t)$
Plant Type	Plant Health	Water Amount	Seed Locations	Plant Type Distribution	Total Canopy Coverage	Garden Diversity	Garden Sustainability

TABLE I - Simulator local and global time-dependent state variables



(a) Light Allocation (b) Water Use

Fig. 2. **Light and Irrigation Models.** Each plant receives light based on the size of its unoccluded leaf area in the grid. **Left:** For each plant, the number of grid points visible overhead determines the amount of light it receives, while occluded points receive light in an exponentially decaying fashion. **Right:** The plant then uses water 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 surroundings.

are less than distance r_i away from its center coordinate. The approximate unoccluded leaf area is then simply the number of coordinates L_i which do not belong to L_j, for any other plant j with a taller height. Light is distributed in an exponentially decaying fashion, where the i^{th} tallest plant at $p(x,y)$ receives LD^i amount of light from $p(x,y)$, where LD is a light decay coefficient in the range $(0,1)$.

Next, once each plant has been allocated light, we model the water use of each plant. Here, our assumption is that plant growth potential is proportional to the amount of sunlight they can intercept, but actual growth is dependent on the plants' access to sufficient water resources, allowing for transpiration and gas exchange, and therefore photosynthesis [17]. The amount of water required by a plant therefore depends on the sunlight obtained from the previous step, and by extension the size of the plant itself.

From this assumption, the maximum water amount desired by a plant is modeled as:

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

where L_u is the unoccluded leaf area of the plant, equivalent to the maximum amount of sunlight it can receive, and c_1 and c_2 are plant-specific parameters that adjust its growth pattern. Larger values for c_1 correspond to higher water use efficiency, i.e. higher biomass accumulation per unit of water. Larger values for c_2 correspond to higher overall productivity, i.e. a higher attainable growth at each timestep [18].

While the simulator uses a grid-based model, it draws from the zone-of-influence approach described by [10], allowing a plant to only uptake water from a circular zone defined by its radius, i.e. from coordinate points that are within radius distance of the plant's central coordinate. To model the sharing of water resources by plants in the same general area, Firbank and Watkinson [19] used relative sizes to allocate proportions of water to each plant. We take a similar approach, but add randomness in order to simulate more aggressive competition. At each coordinate, we iterate in a random order through the plants whose leaf areas cover that coordinate. For each such plant, we allow it to use the maximum amount of water it desires from this coordinate, defined as w_{max} divided by its total approximate leaf area, provided there is still water remaining. Thus, plants will still take a proportion of the water in their ecological neighborhood in expectation, but the randomness allows for some plants to take more water than usual and dominate their surrounding competition. Furthermore, 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.

Once the water at each coordinate has been allocated, we compute a growth value for each plant, defined as:

$$G = c_1 \cdot \min\left(w, w_{max}\right)$$

where w is the actual amount of water this plant was able to uptake. Note that the water allocation process enforces that a plant obtains no more than $w_{max} = \frac{c_2}{c_1}\sqrt{L_u}$ units of water in the best case, so this is essentially $G = c_1 w$, a linear function of the water used.

Each plant must divide its growth value G between vertical growth (increasing its height) and radial growth (increasing its radius). We make this assumption because plants are inherently phototropic and adjust their growth to seek light and escape shade [20], thus they should divide G strategically to ensure maximum unoccluded leaf area. There is a tradeoff: plants will typically seek to increase their radius as much as possible, but if their leaves are occluded by taller plants, they will increase their height in order to outgrow competitors and gain an advantage. We define $l_{o,i}$ and $l_{u,i}$, the number of points where plant i is occluded and unoccluded respectively, and model this dynamic as follows:

$$\begin{aligned}
l_p &= \frac{l_{u,i}}{l_{u,i} + l_{o,i}} \\
r_i &= \max(k_1, \min(l_p, k_2)) \\
G_{radial} &= r_i G \\
G_{vertical} &= (1 - r_i) G
\end{aligned}$$

where k_1 and k_2 are plant-specific constants indicating lower and upper bounds on the proportion of G the plant is willing to allocate to purely radial growth. This is reflective of the

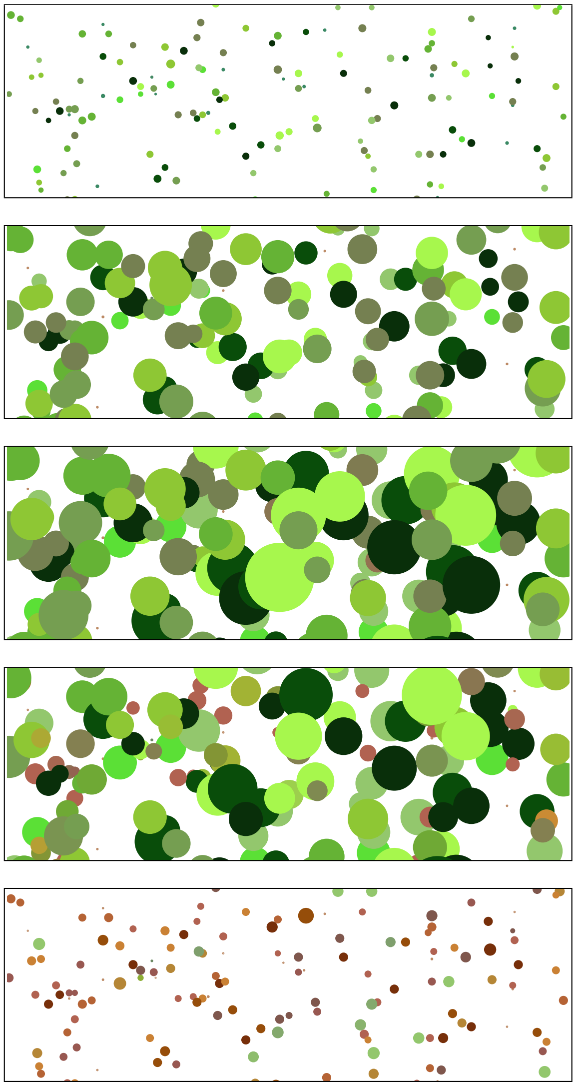


Fig. 3. **Plant Life Stages**. Each plant is modeled with a life cycle trajectory, consisting of five stages (from top to bottom image): germination, vegetative, reproductive, senescence, and death. When plants get underwatered or overwatered, their radius decays exponentially and their color turns brown, 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.

genetically ingrained habit and morphology of the individual species. For instance, higher values of k_1 reflect a rosette habit, in which very little vertical growth occurs between individual leaves until the transition to reproduction [21]. We estimate k_1 and k_2 by taking a ratio of height and radius measurements collected from each plant species in an experimental real garden, as can be seen in the project's open-access repository. In the future, we intend to tune these parameters with more data.

Finally, we increment the radius and height of each plant according to the computed radial and vertical growth values, G_{radial} and $G_{vertical}$.

B. Life Cycle

Each plant is modeled with a biologically standard life cycle trajectory, consisting of five stages: germination, vegetative, reproductive, senescence, and death [22], [23]. Plants progress from one stage to the next after a number of timesteps sampled from a discretized Gaussian distribution, specific to the plant type, as the model assumes that plants of the same type share transition times between stages [24].

1) Germination: When a seed is planted, it begins the germination phase. During this phase, it only occupies the single coordinate point where it was planted, and has both 0 radius and height. It uses light and water according to the model specified in IV.A, but does not change in radius or height. When the plant transitions to the next stage, it gains a nonzero radius and height, each sampled from a plant type specific Gaussian distribution.

2) Vegetative: During the vegetative phase, the plant obtains resources and grows according to the model specified in IV.A, unless it experiences stress from over or underwatering, as further detailed in IV.C.

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

4) Senescence: During the senescence phase, the plant obtains resources according to the model specified in IV.A, but its desired water amount is reduced to a proportion of its usual desired water amount, which linearly decreases to 0 over time. More specifically, if w_d is its desired water amount, should it have been in the vegetative or reproductive stage, then its adjusted desired water amount, $\widetilde{w}_d$, is calculated as:

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

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. However, the plant's accumulated resources no longer go towards its growth. Instead, the radius of the plant decays exponentially to a proportion of its radius at the beginning of the stage. This proportion is a parameter associated with the plant type. Plant height does not change during this phase.

5) Death: When the plant dies, it no longer uses resources nor changes in radius or height. However, it continues to occupy space in the garden, potentially occluding plants.

Fig. 3 shows snapshots from a simulated garden at every stage, where the top image corresponds with the germination phase and the bottom corresponds with the death phase.

C. Water Stress

To model the detrimental effects of suboptimal irrigation, the plant model includes a specific response to over and underwatering when in the vegetative and reproductive stages. While in these stages, a plant is considered overwatered if the total amount of soil moisture within its radius, $w(t)$, exceeds its desired water amount, w_d. More concretely, a plant is overwatered if:

$$w(t) \geq T_o w_d$$

where T_o is a threshold parameter associated with plant type. Similarly, a plant is considered underwatered if the amount of water it uptakes, $\tilde{w}(t)$, falls below some proportion of its desired water amount:

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

where T_u is also a plant type specific threshold parameter.

While the plant remains in either stressed state, its radius decays exponentially to some fraction of its prior radius, similar to its behavior in the senescence phase. Furthermore, we visualize the effects of stress on plants by modifying their color to become progressively more brown as they continue to be stressed. If a plant remains in either stress state after a number of timesteps, then the plant immediately moves to the death stage. If a previously stressed plant is no longer stressed, then for the current timestep the plant's radius will decay, but it will return to its normal behavior the next timestep. Afterwards, the plant will continue its normal behavior with no lingering effects. The duration of the plant's stage is not affected by its stress during the stage. The plant's stress state persists when it transitions from the vegetative stage to the reproductive stage.

D. Irrigation

For each discrete point in AlphaGardenSim, soil moisture dynamics are independently simulated, without interactions between different spatial locations. One-dimensional soil moisture dynamics can be modeled according to the Richards equation [25], which takes the form of the following nonlinear partial differential equation:

$$\frac{\partial s}{\partial t} = -\frac{\partial q}{\partial z} - S(h)$$

where s is the volumetric water content in the soil, t is time, $\frac{\partial q}{\partial z}$ is a term that describes the flow of soil water as a result of differences in soil water potential, and $S(h)$ is a term that describes the soil water extraction by plant roots.

Soil moisture dynamics for each spatial point $p(x, y)$ is based on the following discrete-time linear approximation of this differential equation, proposed by Tseng et al. [26]:

$$w(x, y, t) = w(x, y, t-1) - d + a(x, y, t) - U(x, y, t)$$

where $w(x, y, t)$ is the soil moisture content at point $p(x, y)$ during time t, $a(x, y, t)$ is the amount of irrigation applied, and $U(x, y, t)$ is the plant water uptake rate. d specifies local water loss at each point, and accounts for all contributing factors, such as soil drainage and evaporation. In AlphaGardenSim, d is approximated by a constant quantity uniform for each point in the garden, $U(x, y, t)$ is accounted for by the stochastic plant water uptake model, and $a(x, y, t)$ is specified by an irrigation policy.

Soil moisture levels are initialized according to an i.i.d. Gaussian distribution $N(\mu, \sigma^2)$ for each point in the garden at the start of each episode, with an irrigation amount also specified for each point. Additionally, we enforce that soil moisture content remains non-negative throughout the episode. Thus, soil moisture dynamics can be described by the following equations:

$$w(x, y, 0) \sim \max(N(\mu, \sigma^2), 0) + a(x, y, 0)$$

$$w(x, y, t) = \max(w(x, y, t-1) - d + a(x, y, t) - U(x, y, t), 0)$$

E. Diversity

Plants in a polyculture garden must be pruned to prevent invasive species from dominating. If an invasive plant's growth is too aggressive in relation to other plants, it uses more resources and hinders their growth. We seek to diversify the garden through pruning. We model diversity as the entropy of the global population in the garden:

$$v(t) = H\left(\mathbf{p}\left(k, t\right)\right)$$

We normalize the entropy, so that when $\mathbf{p}(k, t)$ is uniform, $\bar{H}\left(\mathbf{p}\left(k, t\right)\right)$ is 100%, whereas if $\mathbf{p}(k, t)$ is completely unbalanced, for instance $\mathbf{p}(k, t) = [1, 0, 0, ..., 0]^T$, then its normalized entropy is 0%. Since $\mathbf{p}(k, t)$ represents the leaf coverage ratio per plant type, if $\mathbf{p}(k, t)$ is uniform, the garden is maximally diversified, and vice versa. To increase the diversity in the garden, an agent can prune plants that spread more than others. In AlphaGardenSim, pruning reduces the radius of the pruned plant by a fixed percentage.

V. Experiments

We evaluate the performance of different polyculture farming automation policies by assessing their ability to reduce water use and maximize plant diversity in AlphaGardenSim, and we explore the robustness of the policies to varying garden settings, some of which include invasive species. In our experiments, we use a set of 13 edible plant types with different germination times, maturation times and growth rates, sampled from plant-specific Gaussian distributions, as specified in the project's open-access repository.

At the beginning of the simulation, n points are randomly picked and planted with seeds of k types. The seed locations $\mathbf{s}(x, y) = \mathbf{d}(x, y, 0)$ are saved and remain fixed during the entire simulation period. As plants grow and accumulate resources from the garden, each plant's health level $h(x, y, t)$ is tracked, determined by its water-stress level in the previous days and the current amount of water accessible at the plant's location (see IV.C).

The state of the garden at each point $p(x, y)$ is defined as the tuple of all local and global quantities (see Table I):

$$\mathbf{S}(x, y, t) = (\mathbf{d}(x, y, t), w(x, y, t), h(x, y, t), \mathbf{p}(k, t), v(t), s(t))$$

To simulate sensor precision limitations, we define $\tilde{\mathbf{S}}(x, y, t)$, a sector of size $\frac{M}{10} \times \frac{N}{10}$ centered at $p(x, y)$ representing the area observable at time t. In this setting, a policy's access to local quantities $\mathbf{d}(x, y, t)$, $w(x, y, t)$ and $h(x, y, t)$ is limited to points within the sector's boundaries. In addition, we define A, the set of actions an agent can apply; watering, pruning, both, or none. For each observation, an agent chooses $a(x, y, t) \in A$, the action applied at point $p(x, y)$ at time t.

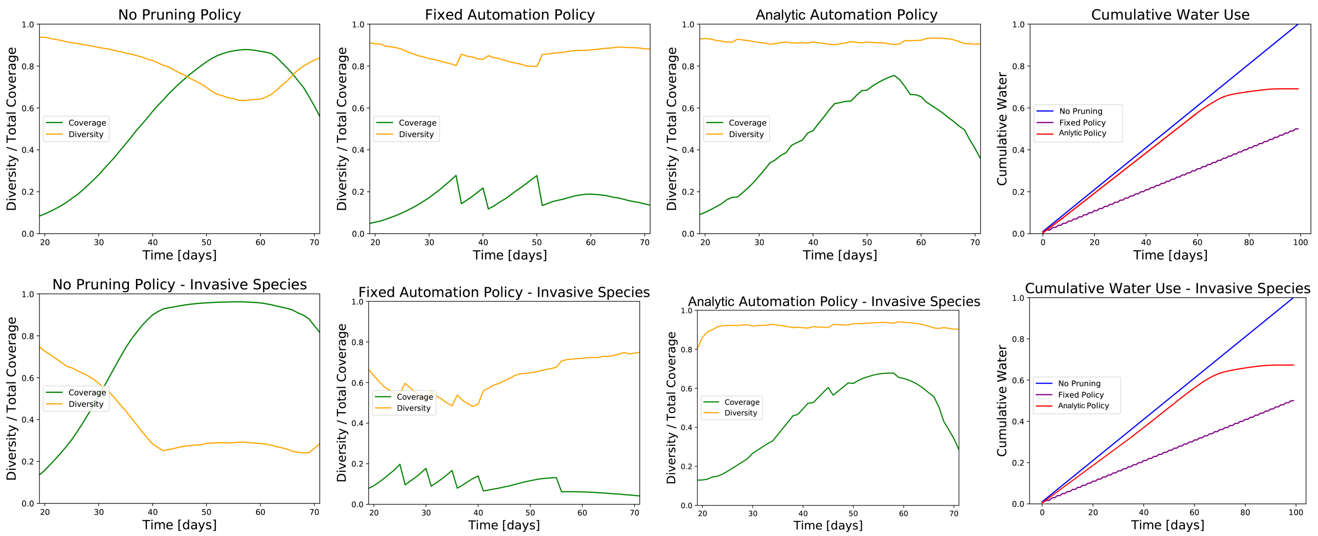


Fig. 4. **Simulation Experiments.** The total plant coverage and diversity in a garden over a period of 50 days out of a 100 day growth period, as a result of using two different automation policies and a policy that does not prune to provide a baseline for growth and diversity. We focus on the period between days 20 and 70 because prior to day 20 the policies do not prune, and after day 70 plants start to die. We can see that there is a trade-off between the coverage and diversity. **Top row:** each policy acts on a general garden. With no pruning, the coverage in the garden is high while the diversity is low. The fixed policy manages to increase the diversity, at the price of reducing the coverage dramatically. The analytic policy finds the balance between the two metrics and achieves high diversity and coverage. **Bottom row:** the policies act on gardens that contain invasive species. If not pruned, the invasive species spreads rapidly until it covers the majority of the garden. The fixed policy tries to slow the spread but fails since it prunes both the invasive species and the other plants, and the invasive species spreads between pruning intervals. The analytic automation policy achieves worse performance than before, but manages to keep a balanced garden by pruning areas covered by the invasive species more frequently.

A. *Experimental Setup*

For experiments, we initialize a 150×300 garden with m plants, each sampled with replacement from k plant types and seeded at points $s(x, y)$, randomly sampled from all garden points. At each time step in the garden, we sample m sectors centered at each $s(x, y)$, as well as an additional $\frac{m}{10}$ sectors centered at non-seed points $p(x, y)$. One hundred garden days are simulated. For light distribution, we choose a light decay LD of 0.5.

We allow each action $a(x, y, t)$ to affect a small window in the center of a sector. During irrigation, we update each $w(x, y, t)$ of an 11×11 square centered around $s(x, y)$ as detailed in IV.D with $a(x, y, t) = 1$. For pruning, we create a 5×5 window centered around $s(x, y)$ and collect all plants that are visible inside the window from overhead. Then we prune each plant, reducing each of their radii by 15%. Pruning actions in the simulator are only applied after a specified number of days, to allow plants to germinate and grow. We set this prune delay to 20 days, after most plants reach the vegetative stage.

Plant overwatering and underwatering thresholds are closely related to the plant's neighborhood. An overwatered threshold of $T_o = 100$ was chosen as it is the maximum amount of water in the 10×10 square around the plant. We chose such a square to promote initial plant growth in our experiments. We set the underwatered threshold to $T_u = 0.1$.

At each time step, the sectors are sampled in a random order, with the policy acting on each observation independently. Once all actions are accumulated, we update the garden's water grid with irrigation amounts and update pruned plant radii. Afterwards, we distribute light, allow plants to use water and grow, and simulate water drainage and evaporation.

B. *Policies*

We implement two policies: a fixed policy that irrigates according to a fixed schedule and prunes uniformly, and an analytic policy that adapts according to the plants' needs and the garden's current diversity.

1) Fixed Policy: The fixed policy acts according to a fixed irrigation schedule, irrigating each sector it observes every two days. This method resembles the one applied in many farms and greenhouses, where an array of drippers or sprinklers irrigates in fixed time intervals at fixed locations. In addition, every 5 days, the policy prunes all the plants that grew beyond a fixed radius, limiting plants from over-spreading. To prevent overpruning, we adjusted pruning to reduce a plant's radius by 5%.

2) Analytic Policy: The analytic policy utilizes soil moisture and plant health to dynamically prune and irrigate each sector it observes. The policy chooses among four actions: irrigate, prune, irrigate and prune, or do nothing. The policy independently decides whether to irrigate and whether to prune, and these decisions are combined to produce an action.

If a sector contains no plants or only dying plants, the policy does not irrigate. If there are any underwatered plants inside the irrigation window, irrigation is selected. Otherwise, we sum $w(x, y, t)$ for all $p(x, y)$ in $\tilde{\mathbf{S}}$. To account for overwatering, wherever $h(x, y, t)$ indicates an overwatered plant, we double $w(x,y,t)$ in the summation. If the final sum is less than a threshold, the policy irrigates.

To decide when to prune, the policy normalizes $\mathbf{p}(k,t)$. If any plant type inside the pruning window has normalized coverage greater than a threshold, the policy will choose to prune.

C. Evaluation

We evaluate each policy on 20 randomly seeded experiments per setting, using the following metrics:

1) Total Plant Coverage - We sum the total 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*:

$$TC = \sum_{k \in (K \setminus soil), t} \mathbf{p}(k,t)$$

2) Average Diversity - We average the diversity in a single experiment, ignoring the first 20 days and last 30 days of the growing period (i.e. averaging between days 20 and 70 of the growing period, $T = 50$):

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

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

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

We conduct a first set of experiments on a general setting consisting of 200 plants from 10 types, for a growing period of 100 days. This setting represents a general and unbiased setup where there is a large variety of plants growing in close proximity. This leads to increased competition for resources, and similar initial conditions for each plant in expectation. We evaluate and compare the performance of the two automation policies, and an additional policy that provides maximal irrigation and does not prune as a baseline. Averaged over 20 test gardens, the results are summarized in Table II. The analytic policy balances pruning and irrigation and achieves higher diversity with respect to the other policies. The top row in Fig. 4 shows the total coverage and diversity in a garden during a single experiment, and provides intuition for the performance of each automation policy; when irrigation is not limited and the garden is not pruned, the total coverage reaches its maximal peak, but the diversity is low due to the competition between plants. A fixed automation policy manages to maintain diversity to some extent, at the cost of overpruning. The analytic automation policy maintains high diversity, but seems to be limited by slow growing plants, which cause it to heavily prune faster growing plants. Nevertheless, it achieves high coverage and diversity. It also provides sufficient irrigation, while the fixed policy underwaters and the no-pruning baseline overwaters.

Policy	Coverage	Diversity	Water Use
No Pruning	**30.13**	0.78	100.00
Fixed	9.27	0.87	**50.00**
Analytic	19.91	**0.91**	67.80
No Pruning w/ Invasive	**37.30**	0.49	100.00
Fixed w/ Invasive	6.64	0.72	**50.00**
Analytic w/ Invasive	17.00	**0.89**	67.21

TABLE II - Policy evaluations averaged over 20 test gardens with and without invasive species. Coverage is the sum over 50 days of percentage of garden covered at each day, diversity is the average of the normalized entropy of leaf distribution over 50 days, and water use is the sum of the water applied over 100 days. Results show a tradeoff between maximizing yield and maximizing diversity, and suggest that a tuned analytic policy can balance those to achieve gardens with high coverage and diversity.

To examine the effects of an invasive species in the garden, we conduct a second set of experiments in which we replace one of the plant types with an invasive species. Invasive species grow rapidly and consume the resources in the garden. To simulate an invasive species, we create a plant with a short germination period, long vegetative and reproductive periods, and large maximal radius. Fig. 5 shows a comparison between the growth of a garden with and without an invasive species, marked with a red color, during 3 phases: germination, the reproductive phase, and after most plants have died. From the second row of Fig. 4 we can see that if not pruned, the invasive species has a negative impact on diversity. The invasive species spreads rapidly until it covers the majority of the garden, and stays dominant during a long reproductive phase, preventing other plants from growing. The fixed automation policy struggles to hold back the spread, as after each pruning, the invasive species recovers and continues to spread. The analytic automation policy performs better, increasing the pruning rate where there are invasive plants, and reducing it where there are none, giving other plant types the chance to grow.

VI. Conclusion

We present AlphaGardenSim, a fast, first order simulator that uses plant growth models to simulate inter-plant dynamics and competition for light and water. We use it to tune an analytic automation policy that achieves high coverage and diversity, even in the presence of invasive species. A key factor AlphaGardenSim does not model is the effects of companion plants, that can assist neighboring plants by repelling pests or providing nutrients and shade. In future work, we will further tune the simulator with real world measurements. We will then apply the simulator to learning policies that optimize leaf coverage and diversity, and evaluate the performance of the resulting policies on real gardens.

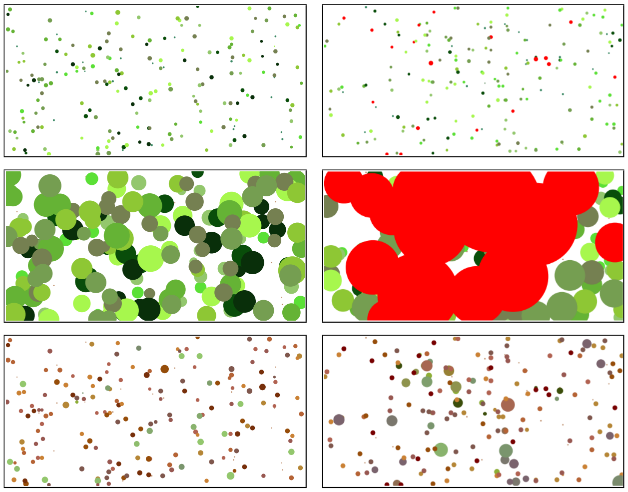


Fig. 5. Invasive species grow rapidly and spread, consuming the resources in the garden and preventing the growth of other plants. We see the comparison between a regular garden (left column) and a garden with an invasive species (right column). **Top row:** Germination phase, both gardens look similar but the invasive species starts to germinate earlier than other plant types. **Middle row:** At the peak of the reproductive phase, the right garden is dominated by the invasive species. **Bottom row:** At the end, all plants die and the gardens reach a similar state.

ACKNOWLEDGEMENTS

This research was performed at the AUTOLAB at UC Berkeley in affiliation with the Berkeley AI Research (BAIR) Lab, Berkeley Deep Drive (BDD), the Real-Time Intelligent Secure Execution (RISE) Lab, and the CITRIS "People and Robots" (CPAR) Initiative. This research was supported in part by the RAPID: Robot-Assisted Precision Irrigation Delivery Project (USDA 2017-67021-25925 under NSF National Robotics Initiative). The authors were supported in part by donations from Siemens, Google, Toyota Research Institute, Honda Research, and Hewlett-Packard.

We thank our 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, Sebastian Oehme, Anna Deza, Raven Huang and Atsunobu Kotani.

REFERENCES

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

[2] M. Liebman, "Polyculture cropping systems," in *Agroecology*. 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, 2018.

[4] J. W. Jones, G. Hoogenboom, C. H. Porter, K. J. Boote, W. D. Batchelor, L. Hunt, P. W. Wilkens, U. Singh, A. J. Gijsman, and J. T. Ritchie, "The dssat cropping system model," *European journal of agronomy*, vol. 18, no. 3-4, pp. 235–265, 2003.

[5] 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," *Agronomy Journal*, vol. 101, no. 3, pp. 426–437, 2009.

[6] 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," *Environmental modelling & software*, vol. 50, pp. 25–36, 2013.

[7] N. Chebrolu, T. Läbe, and C. Stachniss, "Spatio-temporal non-rigid registration of 3d point clouds of plants."

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

[9] W. J. Price, B. Shafii, and D. C. Thill, "An individual-plant growth simulation model for quantifying plant competition," 1994.

[10] 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," *Perspectives in Plant Ecology, Evolution and Systematics*, vol. 9, no. 3-4, pp. 121–135, 2008.

[11] 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*. Springer, 1990, pp. 229–239.

[12] M. Wiggert, L. Amladi, R. Berenstein, S. Carpin, J. Viers, S. Vougioukas, and K. Goldberg, "Rapid-molt: A meso-scale, open-source, low-cost testbed for robot assisted precision irrigation and delivery," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2019, pp. 1489–1496.

[13] 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)," 2017.

[14] E. Rohmer, S. P. Singh, and M. Freese, "Coppeliasim (formerly v-rep): a versatile and scalable robot simulation framework," in *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[15] K. R. Aravind and P. Raja, "Design and simulation of crop monitoring robot for green house," 2016.

[16] H. P. Koelewijn, "Rapid change in relative growth rate between the vegetative and reproductive stage of the life cycle in plantago coronopus," *New phytologist*, vol. 163, no. 1, pp. 67–76, 2004.

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

[18] T. C. Hsiao, "Effects of drought and elevated co 2 on plant water use efficiency and productivity," in *Interacting stresses on plants in a changing climate*. Springer, 1993, pp. 435–465.

[19] L. Firbank and A. Watkinson, "A model of interference within plant monocultures," *Journal of Theoretical Biology*, vol. 116, no. 2, pp. 291–311, 1985.

[20] A. Goyal, E. Karayekov, V. C. Galvão, H. Ren, J. J. Casal, and C. Fankhauser, "Shade promotes phototropism through phytochrome b-controlled auxin production," *Current Biology*, vol. 26, no. 24, pp. 3280–3287, 2016.

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

[22] A. Zardilis, A. Hume, and A. J. Millar, "A multi-model framework for the arabidopsis life cycle," *Journal of experimental botany*, vol. 70, no. 9, pp. 2463–2477, 2019.

[23] T. Hirose, T. Kinugasa, and Y. Shitaka, "Time of flowering, costs of reproduction, and reproductive output in annuals," in *Reproductive allocation in plants*. Elsevier, 2005, pp. 159–188.

[24] P. J. Lumsden and A. J. Millar, *Biological rhythms and photoperiodism in plants*. Bios Scientific Publishers, 1998.

[25] J. C. Van Dam, J. Huygen, J. Wesseling, R. Feddes, P. Kabat, P. Van Walsum, P. Groenendijk, and C. Van Diepen, "Theory of swap version 2.0; simulation of water flow, solute transport and plant growth in the soil-water-atmosphere-plant environment," DLO Winand Staring Centre, Tech. Rep., 1997.

[26] D. Tseng, D. Wang, C. Chen, L. Miller, W. Song, J. Viers, S. Vougioukas, S. Carpin, J. A. Ojea, and K. Goldberg, "Towards automating precision irrigation: Deep learning to infer local soil moisture conditions from synthetic aerial agricultural images," in *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2018, pp. 284–291.