AlphaGarden: Learning to Autonomously Tend a Polyculture Garden

Mark Presten[1], Yahav Avigal[1], Mark Theis[1], Satvik Sharma[1], Rishi Parikh[1]
Shrey Aeron[1], Sandeep Mukherjee[1], Sebastian Oehme[2], Simeon Adebola[1]
Walter Teitelbaum[3], Varun Kamat[1], and Ken Goldberg[1]

***Abstract*— This paper presents AlphaGarden: an autonomous polyculture garden that prunes and irrigates living plants in a $1.5m \times 3.0m$ physical testbed. AlphaGarden uses an overhead camera and sensors to track the plant distribution and soil moisture. We model individual plant growth and interplant dynamics to train a policy that chooses actions to maximize leaf coverage and diversity. For autonomous pruning, AlphaGarden uses two custom-designed pruning tools and a trained neural network to detect prune points. We present results for four 60-day garden cycles. Results suggest AlphaGarden can autonomously achieve 0.96 normalized diversity with pruning shears while maintaining an average canopy coverage of 0.86 during the peak of the cycle. Code, datasets, and supplemental material can be found at https://github.com/BerkeleyAutomation/AlphaGarden.**

I. Introduction

Industrial agriculture is based on monoculture, where a single crop type is cultivated, requiring substantial use of fertilizer, pesticides, and water [1], [2]. Polyculture farming, on the other hand, is a sustainable practice in which multiple crop types are intermixed. The inherent benefits of polyculture farming include reduced weeds and soil erosion, better resistance to pests and viruses, and increased water and soil nutrient efficiency [3], [4], [5], [6]. However, polyculture farming is more laborious than monoculture, requiring maintenance to ensure that larger, more dominant plant types do not overwhelm smaller, slow-growing plants.

We present AlphaGarden, an implemented mechanical and algorithmic robotic system for autonomous polyculture plant cultivation that applies policies learned in simulation to tend to a physical garden (see Fig. 1). Autonomous policies control watering and pruning actions. We introduce two novel pruning tools and algorithms that can autonomously prune plants. Experiments suggest that AlphaGarden is capable of pruning plants to facilitate plant diversity while maintaining high canopy coverage. AlphaGarden builds on Plant Phenotyping, Bounding Disk Tracking, and AlphaGardenSim, a polcyulture garden simulator, as presented in [7], [8]. To the best of our knowledge, this is the first autonomous system in a polyculture farming setting.

This paper makes 4 contributions:

1) AlphaGarden, a fully autonomous physical polyculture garden testbed with actuators for irrigation and pruning and policies learned from simulation,
2) Significant extensions to AlphaGardenSim [7], [8] to efficiently model growth and phenotyping,
3) The Variable Radius Poisson Disk (VARPOD) algorithm for seed placement, a bounding disk tracking algorithm for plants, and a learned prune point identification network for autonomous pruning,
4) and experimental data from the execution of AlphaGarden over 120 days with analysis of the results.

Fig. 1: **AlphaGarden. Top:** Physical testbed with the FarmBot gantry system. AlphaGarden includes a custom Rotary Pruner, custom Pruning Shears, a depth sensor, an on-board snake inspection camera, an overhead camera, and soil moisture sensors. **Bottom:** Overhead image of growth based on two mirrored seed placements for Garden Cycles 3 and 4.

II. Related Work

Cultivating plants has been an essential human activity for over 10,000 years. Humans have continuously improved farming techniques, and in recent years, have introduced methods for agricultural automation. In 1995, the Telegarden, an art installation by Goldberg et al. [9], [10], allowed internet visitors to interact with a remote garden by planting and watering plants. Wiggert et al. [11] created a testbed for monitoring plant growth and water stress in real time. Correll et al. [12] designed a distributed autonomous gardening system with mobile manipulators that detect plants, irrigate, and grasp fruits. We extend these prior works by creating a fully autonomous polyculture garden.

Existing plant simulators model growth of single species [13]. Examples such as DSSAT [14] and AquaCrop [15] simulate a growth period in large-scale monoculture farms. AlphaGardenSim [7], [8] simulates a

[1]The AUTOLab at UC Berkeley (automation.berkeley.edu) {mpresten, goldberg} @berkeley.edu [2]Department of Electrical and Computer Engineering, TU Munich sebastian.oehme@tum.de [3] Department of Robotics Engineering, UC Santa Cruz wteitelb@ucsc.edu

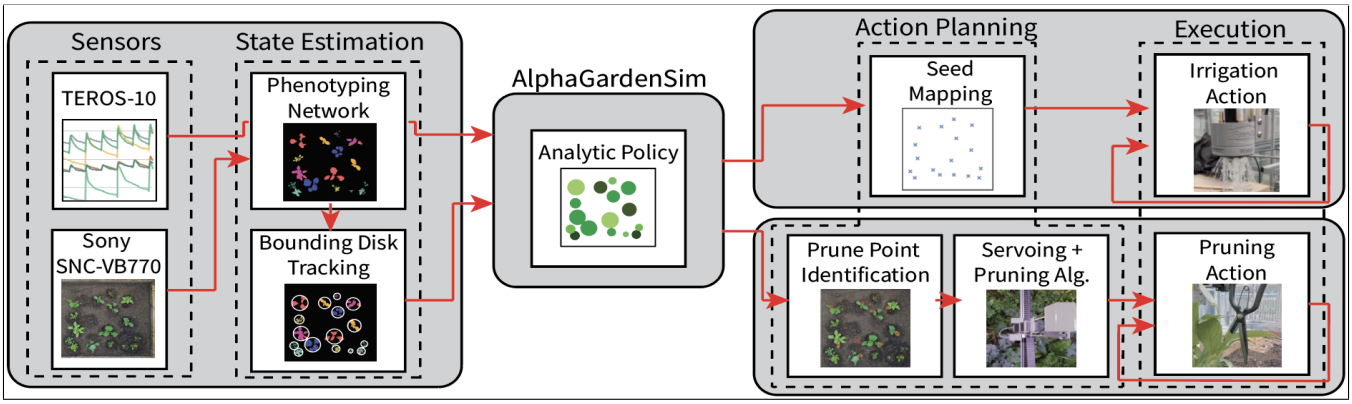

Fig. 2: **AlphaGarden Autonomous Pipeline.** The leftmost box encompasses the Real2Sim phase, while rightmost shows Sim2Real. The overhead Sony camera and TEROS-10 soil moisture sensors gather data. This is followed by a state estimation process to identify individual plants. AlphaGardenSim determines appropriate actions in real time, which is followed by an action planning phase and execution on the physical garden.

polyculture garden using first order models of plant growth with inter-plant dynamics and competition for water and light. The model parameters used in AlphaGardenSim were tuned using real-world measurements, including parameters for plant growth, plant companionship, and water dynamics.

Phenotyping is an important task for monitoring plants, similar to object tracking and identification. Ayalew et al. [16] present a method to use an unsupervised domain adaptation network to adapt the meticulously pre-labeled Computer Vision Problems in Plant Phenotyping (CVPPP) dataset [17], [18] to other plant and image domains. The data consists of single plants, their leaves, and a point map of leaf centers. This reduces human effort required to track, count, and identify leaf centers.

Pruning is a necessary capability to tend a polyculture garden. Previous work in autonomous pruning includes rose and bush trimming with a robot arm [19], [20]. Habibie et al. [21] trained a Simultaneous Localization and Mapping (SLAM) algorithm to enable automated fruit harvesting in a red apple tree field. Cuevas-Velasquez et al. [19] demonstrated success using visual servoing to account for changes in stem poses to determine cutting points. In a controlled greenhouse, Van Henten et al. [22] used a robot with a thermal cutting tool to harvest cucumbers. We extend previous work by developing an autonomous pruning pipeline for trimming leaves in a controlled environment.

FarmBot is an open source gantry robot commercially available since 2016 that is used in AlphaGarden. Previous work with this system has examined kinematic modeling to enhance FarmBot trajectory planning [23]. A team from Telkom University used FarmBot to automate seed placement, watering policy, and plant monitoring routines [24]. More recently, researchers have proposed a FarmBot simulator "to support the development of a control software able to implement different [precision agriculture] strategies" [25].

III. The Polyculture Growing Problem

A Garden Cycle consists of planting an arrangement of selected plant types, then irrigating and pruning until growth is completed. Garden quality is a function of coverage, plant diversity, and water usage. In this paper, we focus on maximizing canopy coverage and diversity.

Given $i > 5$ plant types with estimated germination time g_i, maturation time m_i, and maximum radius R_i, a Garden Cycle is defined by a germination stage, a growth stage $l_i = m_i - g_i$, a waiting stage, and a wilting stage, as defined in [7].

Given a rectangular garden of dimensions (w, h), we define coverage, $c(t)$, as the sum of all plant type canopy coverage, $\Sigma_i c_i(t)$, over total area $w \cdot h$ at day t.

For each plant type i with maximum radius R_i, we define garden diversity as follows:

$$d(t) = H(c_i(t) \cdot (R/R_i)^2)$$

where $H(\cdot)$ is an entropy function and R is the average maximum radius over all plant types. Multiplying by $(R/R_i)^2$ normalizes each plant type's canopy coverage. Overall, the AlphaGarden system computes a seed arrangement, which are manually planted, then observes growth from an overhead camera, and monitors soil moisture. As seen in Fig. 2, AlphaGarden uses a learned policy π to choose irrigation and pruning actions to execute autonomously.

IV. Hardware

a) FarmBot: AlphaGarden is a $3.0m \times 1.5m$ raised planter bed located in the UC Berkeley greenhouse. A commercial FarmBot [26] is installed over the planter bed frame. This CNC robot may travel to any location in the garden from the soil level to $0.4m$ above. The FarmBot also features a magnetic universal tool mount (UTM) on its Z-axis that can automatically swap between tools stored on the west side of the bed.

b) Water Nozzle: To build a functional autonomous garden, we designed and fabricated custom tools for watering and pruning. The watering nozzle creates a wide and uniform output stream [8], which leads to better soil water retention and less erosion. The watering nozzle attaches to the UTM.

c) Rotary Pruner: We developed a custom pruning tool for the FarmBot that utilizes thin, flexible blades rotating at high speeds to cut plants in a 'weed wacker' fashion. We selected an SM Tech 775 Brushed 24V DC motor capable of 12000 rpm to achieve this function. We also designed a

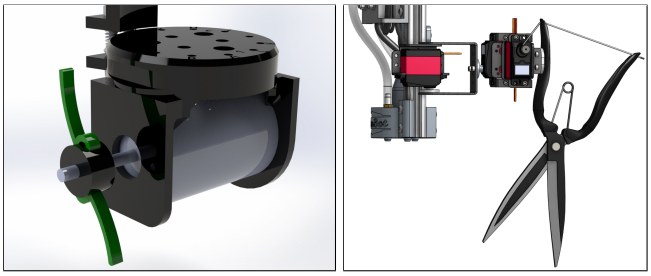

Fig. 3: **AlphaGarden pruning tools. Left:** Rotary Pruner with a high speed motor and trimming blades. **Right:** Pruning Shears with three servos to control closing, tilt, and orientation.

motor housing that inter-operates with the FarmBot UTM. The electrical control includes a relay circuit that governs motor power and uses GPIO to integrate with the FarmBot OS. The FarmBot does not rotate along the Z-axis, so we designed two rotary pruning tools with different orientations: one that cuts along the X-axis and the other along the Y-axis.

d) Pruning Shears: To evaluate another pruning approach, we designed a shear-based pruning tool. A pair of Niwaki Topiary Clippers [27] operate using a steel cable that winds up a spool attached to a YANSHON Digital 360° servo motor. This pruning mechanism is mounted to a 2-axis servo gimbal (using BETU Digital 270° servo motors). The gimbal positions the shears vertically, horizontally, or at any intermediate angle, allowing the FarmBot to trim the top and sides of plants. The servo motors connect to the FarmBot PWM header to integrate with the FarmBot OS.

e) Z-Axis Sensors: A snake inspection camera [28] is located adjacent to the UTM on the Z-axis. It allows for close-up images of plants and soil. We also mounted a Sharp infrared distance sensor [29] to measure the height of plants. Both integrate with the FarmBot OS.

f) Overhead Camera: We mounted a Sony SNC-VB770 digital camera [30] with a $20mm$ Sony lens [31] $2m$ above the garden bed to monitor AlphaGarden. The camera's major requirements include (1) resolution, (2) image distortion, (3) power delivery, and (4) remote data accessibility. The VB770 satisfies these needs. It has a DSLM $35mm$ sensor with a maximum 4240×2832 resolution (1.4x higher than 4K) image mode. The $20mm$ lens minimizes distortion and allows us to capture the entire garden.

g) Soil Moisture: To measure soil moisture and model soil dynamics, we distributed six TEROS-10 soil moisture sensors [32] throughout AlphaGarden. These measure the Volumetric Water Content (VWC) of the soil with a $430mL$ volume of influence. The sensors connect to a ZL6 Data Logger [33], which publishes readings every 30 minutes.

V. Seed Placement Algorithm

The seed placement algorithm described in prior AlphaGarden work [8] has undesirable properties such as clustering plants close together, planting near borders, and allowing for overlap between plants of the same type. This leads to over-competition, under-utilization of garden space, and plant segmentation difficulties.

We present Variable Radius Poisson Disk (VARPOD), a new seed placement algorithm that addresses these issues.

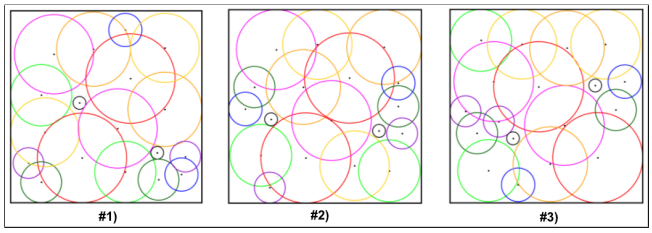

Fig. 4: **VARPOD Seed Placement.** Three seed placements generated by the VARPOD algorithm. The plants are colored by plant type; the voids are colored black. We chose to plant seed placement #1 for garden cycles 3 and 4.

The most recent garden cycle contains 2 of each of the 8 different plant types: kale, turnip, borage, swiss chard, radicchio, green lettuce, red lettuce, and cilantro. This cycle also contains two spaces (termed 'voids') where plants are not allowed to be placed, which mitigate overcrowding.

VARPOD discretizes the garden space into a grid: one unit on the grid represents $1cm^2$ of space. The algorithm first places both voids and then places all plants in the garden in descending order of average radius. Once a plant is placed, it does not move.

VARPOD uses the Variable Radius Poisson Disk distribution [34] to ensure each plant of type i must be at least R_i away from other plants. VARPOD ensures that an additional set of constraints are met as well. First, VARPOD ensures that plants of the same type do not overlap, which makes plant identification more reliable. Second, VARPOD ensures that no plant R_i overlaps with the voids, and that the voids are sufficiently spaced apart from each other and from borders. Third, VARPOD ensures that no plant R_i crosses any border.

When placing plant p, the algorithm first eliminates all points that violate any constraints. The algorithm then assigns a weighted probability to all possible remaining points via c_p, a plant's companionship score as described in [8]. Plant locations are selected according to these weights. Thus, the algorithm likely returns a seed placement where plants have high companionship with neighboring plants.

VARPOD produces a different seed placement for each iteration, which takes about 3 seconds on a 2018 MacBook Pro. Sometimes, VARPOD cannot place all plants in the garden without violating constraints, in which case it returns an unviable garden with fewer plants than specified. In our case of 16 plants in a $1.5m \times 1.5m$ garden (half of the test bed), about 1 in 300 seed placements are viable, resulting in an expected runtime of 15 minutes to generate a viable seed placement.

Fig. 4 shows three seed placements from VARPOD. After qualitatively examining these images, we chose seed placement #1 for garden cycles 3 and 4 because of its high companionship score, same-plant separation, and minimal clustering.

VI. Plant Phenotyping

To estimate the garden state, we use semantic segmentation to identify plant types and growth from an overhead image. The image is cropped to 2000×3780 RGB pixels. We

build on previous work [8] where we trained a model using the UNet architecture [35] and ResNet34 [36] backbone.

The network is trained on 6 hand-labeled overhead images from previous garden cycles. Each image is split into 512×512 RGB patches and augmented via shifting and rotating. We extract leaf masks from various stages in the garden and overlay these leaves on top of the existing patches to augment the dataset. To train, we use a 75-25 train-validation split and with a categorical cross entropy loss function over 100 epochs. The model then outputs a $512 \times 512 \times 9$ array corresponding to the likelihood of each of the 8 plant types and a 9th label, "unknown," (which includes soil) for each pixel in image. We encode each label to an RGB pixel to create a prediction image as seen in Fig. 5.

The baseline model had a mean IoU of 0.71 when compared to the ground truth at day 30, where mean IoU is defined as $\sum_{i=0}^{8} IoU(\text{label}_i)/9$. We found that the network often mislabeled a pixel as a plant type outside the plant's expected radius. To account for this, we include the previous day's garden state as an input to the phenotyping algorithm represented via a mask of size $2000 \times 3780 \times 9$. For each element (x, y, i), where i represents plant type, $(x, y, i) = 1$ if (x, y) is outside the previous day's radius for plant i, and ranges from 2 to 4 as the (x, y) location moves closer to the plant center. We then conduct an element-wise multiplication of the arrays to create a new weighted prediction for each pixel.

Overall, these changes resulted in a mean IoU of 0.85 across the 9 labels on day 30. We saw the highest IoU of 0.97 in borage, which is one of the larger plants. Radicchio, which previously had the lowest IoU, had the largest increase from 0.23 to 0.65 when switching to the new network. We saw an increase in mean IoU of 0.08 when testing every 10 days from day 20 to 60.

The phenotyping algorithm sees a decrease in performance towards later days in the cycle due to occlusion or death of plants. The network scored a mean IoU of 0.42 on day 50. In the future, we plan to address this by collecting more data on plants in the waiting and wilting stage.

VII. Bounding Disk Tracking

To compute irrigation and pruning policies, we convert the plant segmentation mask estimates of each plant's center and radius into the circular model used in AlphaGardenSim. We define a plant's center as the average of all pixel locations in the plant's segmentation mask. The radius is defined as the distance from the center to the farthest point on its contour and accounts for plant centers moving over time due to phototrophy [37] and irrigation [38].

We use a breadth-first-search (BFS) based algorithm to track each plant's center and radius. The algorithm is initialized with seed locations and all plant radii at $0cm$. At each timestep, we use AlphaGardenSim and the prior plant radius to calculate a maximum possible radius by simulating a day of plant growth. Given the prior radius, maximum radius, and (constant) minimum radius, the algorithm traverses outwards from the minimum radius. The algorithm stops when less than 10% of the newly traversed pixels are of the correct type or the maximum radius has been achieved. This process repeats each day for each plant. Even when a plant becomes fully occluded, the algorithm handles radial decrease using AlphaGardenSim's tuned wilting parameters.

Fig. 5: **Phenotyping and Bounding Disk Tracking.** 3 images from days 20, 30, and 40 of garden cycle 4. **Top row:** overhead images overlayed with the estimated circles from the Bounding Disk Tracking algorithm. **Bottom row:** the masks created by the Plant Phenotyping network as well as the estimated circles (same as above).

We compare the disk tracking algorithm to a naive baseline which assumes a static center and radius defined by a logistic curve fit to data from previous gardens. To benchmark against this baseline, we calculate the average movement of each plant's estimated center per timestep, the average maximum distance of a plant's BFS circle's center from its seed placement, and the mean absolute error (MAE) of the BFS-computed radii from growth curve-computed radii. The average distance moved per day was $1.3cm$ and the average maximum distance from the seed placement was $17.1cm$. The average MAE between the logistic curve and the BFS value was $6.4cm$. Generally, larger plants had higher variance because they are more likely to become partially occluded. These metrics show that the new method is more robust to non-ideal behavior and pruning.

VIII. Policy Learning

AlphaGardenSim simulates inter-plant dynamics, including light and water competition between plants in close proximity, and approximates growth in a real greenhouse garden at a fraction of natural growth time, enabling policy learning. The estimated state of the garden, which includes plant types, centers, and radii along with an estimated soil moisture grid, is passed into the simulator. Then, the analytic policy demonstrated in [8] decides what pruning and irrigating actions to execute in the physical garden.

Interacting with AlphaGardenSim produces batches of experiences that are utilized for policy learning. To speed up this process, we reduce the time the environment needs to approximate growth by performing vectorized computations. The refactored simulation environment runs at 455x the speed of natural growth and can simulate 60 plants 42x faster than the non-vectorized version from [8].

Fig. 6: **Prune Point Identification.** Example of all plant leaf centers that were identified by the baseline algorithm (left) and the model (right) applied to an overhead AlphaGarden image. The model identifies more usable points with fewer misclassifications (red circles).

IX. Pruning Actions

After AlphaGardenSim decides which plants to prune, AlphaGarden uses a Prune Point Identification neural network followed by a selection process to identify specific leaves to prune. Then, visual servoing positions the FarmBot above the proper leaf. Finally, the robot uses pruning algorithms, specific to the designated pruning tool, to cut the leaf.

1) Prune Point Identification: AlphaGarden identifies the best leaf to cut after a plant is chosen to be pruned. Our baseline approach takes the average of the extrema of the leaf and center of the plant to find the leaf center. However, this is constrained by the reality of plants' physical makeup, which led us to explore plant phenotyping research.

The model generates a heatmap of likely plant leaf centers. From this, a recursive clustering and thresholding algorithm identifies leaf centers. This recovers leaf centers which the model is less certain about $\in [0, 1]$, compared to the initial normalized lower cutoff of 0.3, while filtering noise. This methodology ensures that prune points do not land on other plants or soil. The model, aided by the processing algorithm, is able to identify 32% more leaf centers than the baseline (see Figure 6). The center of mass for the identified points is, on average, 38% closer to the center compared to the baseline. This is beneficial because pruning closer to the center of the plant allows for pruning actions to cut off a greater portion of the leaf.

2) Visual Servoing: AlphaGarden uses the on-board camera to locate prune points in the physical garden bed. First, the on-board camera takes an image at the relative location of the plant based on the seed placement. This image is then localized within the overhead 'global' image by calculating a normalized correlation coefficient between the images. After finding the best match in the overhead global image, the FarmBot is instructed to move along the vector from the current location to the prune point. Then another 'local' image is taken at this new location, and the iterative cycle continues until the FarmBot reaches within 1 cm of the prune point or reaches an iteration limit.

3) Pruning Algorithms: After locating the leaf to prune, we utilize the depth sensor to find the distance to the target leaf. The Rotary Pruner has a fixed offset in the x and y directions depending on its orientation. The Shears require calculating an orthogonal vector to the vector spanning from the center of the plant to the prune point because the Shears cut along that orthogonal vector. Since the Shears move along a semi-circle centered 1.5 cm from the on-board camera, the FarmBot again moves to account for this offset and to ensure the Shears are directly above the prune point. After a pruning tool is directly above the prune point, the tool moves to the correct depth. The Rotary Pruner toggles on before lowering, while the Shears toggle on upon reaching the depth.

Plant Type	Cut	Pruning Shears Results			Rotary Pruner Results		
		Compl.	Precision	Err.	Compl.	Precision	Err.
Eggplant	1	2	0	B	2	0	B
	2	3	0	A	2	1	A
	3	2	0	B	3	1	A
	4	3	1	A	3	1	A
	5a	2	0	C,D	2	0	B
	5b	3	0	C,D	2	0	B
	6a	2	1	D	2	1	B
	6b	-	-	-	2	1	B
BellPepper	1	1	0	B	1	1	D
	2	2	0	C	1	1	B
	3	3	3	A	1	1	B
	4	1	0	B	3	0	A
	5a	1	0	C,D	3	0	A
	5b-c	1	0	C,D	-	-	-
	5d	3	1	C,D	-	-	-
	6	-	-	-	2	1	B

TABLE I: **Isolated Pruning Experiments** for the Rotary Pruner and Pruning Shears. **Key:** *Completeness-* 3: complete cut, 2: partial cut, 1: missed cut. *Precision-* 1: no damage to other leaves, 0: damage to other leaves. *Error Type-* A: No error, B: location, C: depth, D: Other.

X. Experiments

A. Isolated Pruning Experiments

To evaluate the two pruning tools, we ran isolated pruning experiments using two plants types in pots. We chose eggplant for its large leaves comparable to kale, borage, and turnip, and we chose bell pepper for its smaller abundant leaves similar to cilantro and lettuce. We placed a grown potted plant near the midline of the garden bed, took an overhead image, and then passed a manually annotated plant center and prune point into our servoing and pruning algorithms.

For each tool, we made 5-6 prunings on both plant types, observing completeness of the cut, precision of the cut (if any neighboring leaves were harmed in the process), and any error that may have occurred. We found the Rotary Pruner was more likely to complete a cut, but it also tended to over prune. The Shears, on the other hand, generally caused little secondary damage and debris, but they were more prone to incompletely cut a leaf. Table I shows the results.

B. 4 Garden Cycles

To evaluate AlphaGarden holistically, we ran four autonomous cycles over two 60 day periods. We split the garden into two halves and planted identical seed placements ($1.5m \times 1.5m$) on each. Each half was treated as an independent garden cycle. An overhead image taken at 09:00 daily was passed through the phenotyping network and the bounding disk tracking algorithm so that AlphaGardenSim could process the data and chose which plants to water. After day 30, and every five days after, pruning actions took place. An overhead image taken at 19:00 was processed and passed into AlphaGardenSim as before. The image was subsequently

Plant Type	r_{max}	Cycle 1	Cycle 2	% Change
Kale	37	0.158	0.102	-35.44%
Turnip	33	0.085	0.043	-49.41%
Borage	32	0.122	0.076	-37.70%
Swiss Chard	28	0.105	0.102	-2.86%
Arugula	25	0.098	0.121	23.47%
Radichhio	23	0.034	0.059	73.53%
Red Lettuce	20	0.000	0.057	N/A
Cilantro	19	0.062	0.078	25.81%
Green Lettuce	16	0.028	0.095	239.29%
Sorrel	10	0.002	0.031	1450%
DIVERSITY		0.856	**0.970**	13.32%
COVERAGE		**0.924**	0.784	-15.15%

TABLE II: **Plant Type Metrics for Garden Cycles 1 & 2 .** This table shows diversity and coverage for plant types on day 60. The values for Cycle 1 (not pruned) and Cycle 2 (pruned with Rotary Pruner) are calculated via $[c_i(60) * (R/R_i)^2]$ for each plant type (Section III). The goal of pruning is to foster a diverse garden while maintaining a high coverage.

used for prune point identification and selection. The visual servoing and pruning algorithms were then executed on the chosen leaves.

a) Human Intervention: Although the goal is a fully autonomous polyculture garden, some human intervention was required during Growth Cycles 1-4. Although all decisions were made autonomously, seed planting was performed with manual human intervention. A member of the project team (Mark) was present during all pruning actions, which were executed in batches. Human intervention was used to correct robot position when the Farmbot gantry failed to servo to the correct target location, this occurred in 45% of pruning operations. No other human intervention was performed in terms of weeding or irrigation.

b) Garden Cycles 1 and 2: In Cycles 1 and 2, identical seed placements ($1.5m \times 1.5m$) included the 16 plants mentioned in Section V as well as 2 of both arugula and sorrel. The seed placement was generated via a precursor to the VARPOD seed placement algorithm in [8]. The same irrigation procedures were used for both sides of the garden. In Cycle 1 (left half of the garden bed) there were no pruning actions. In Cycle 2 (right half) pruning actions were executed with the Rotary Pruner. Over 6 pruning sessions for Cycle 2, 42 plants were chosen to be pruned across 6 plant types. The system selected turnip and kale plants for pruning as they grew much faster than the other plants. Due to the numerous prunings and the Rotary Pruner's unclean cuts, we see both turnip plants approach their wilting stage by day 60. Canopy coverage and diversity are reported in Table II. As can be seen by comparing the results with and without pruning, it is clear that pruning increases diversity by creating space for smaller plants to develop.

c) Garden Cycles 3 & 4: For garden cycles 3 (left half) and 4 (right half), we planted two identical seed placements ($1.5m \times 1.5m$) generated with the VARPOD seed placement algorithm. This time the planting pattern was mirrored across the centerline. Arugula and sorrel were not used because they were too large and too small respectively.

On both sides, all pruning actions were performed using the Precision Shears. During Cycle 3, 35 plants were chosen for pruning across 6 plant types. During Cycle 4, 38 plants were chosen for Cycle 4 across 7 plant types. Kale and borage (the larger plants) were most commonly selected in both. No plants exhibited signs of wilting or overpruning by day 60.

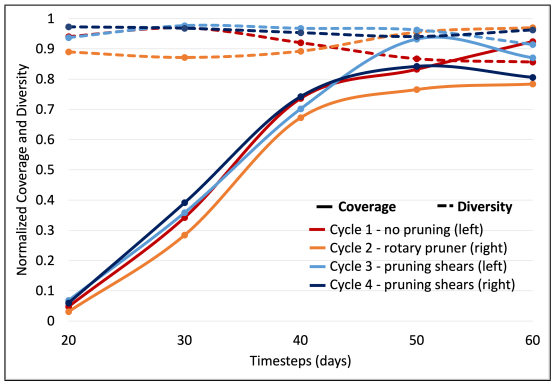

Fig. 7: **Garden Cycle Comparison** using diversity and coverage analysis. Data points were recorded for days 20, 30, 40, 50, and 60.

Fig. 7 compares coverage and diversity for all 4 cycles. For Garden Cycle 1 (no pruning), diversity was significantly lower than the other cycles by day 60. Cycle 3 achieved significantly more coverage during days 45-55.

XI. Conclusion

Despite recent advances in robotics and automation, automating a polyculture garden remains challenging. This paper presents AlphaGarden, an autonomous system that uses learned policies to estimate garden state and to choose pruning actions to increase coverage and diversity. We implemented the AlphaGarden testbed in hardware. We designed and evaluated two custom pruning tools, the VARPOD seed placement algorithm, and Alphagarden over the course of 4 Growth Cycles. In future work, we will evaluate Alphagarden against baselines over future growth cycles and develop algorithms to optimize water usage and further reduce human intervention. All specifications, algorithms, CAD models, and data are available at https://github.com/BerkeleyAutomation/AlphaGarden.

Acknowledgements

This research was performed at the AUTOLAB at UC Berkeley in affiliation with the Berkeley AI Research (BAIR) 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). We thank our colleagues who provided helpful feedback and suggestions, in particular Mary Power, Sarah Newman, Eric Siegel, Isaac Blankensmith, Maya Man, Christiane Paul, Charlie Brummer, Christina Wistrom, and Grady Pierroz.

References

[1] T. S. Rosenstock, D. Liptzin, K. Dzurella, A. Fryjoff-Hung, A. Hollander, V. Jensen, A. King, G. Kourakos, A. McNally, G. S. Pettygrove *et al.*, "Agriculture's contribution to nitrate contamination of californian groundwater (1945–2005)," *Journal of Environmental Quality*, vol. 43, no. 3, pp. 895–907, 2014.

[2] A. Mantovani. (2019) Pesticide risk assessment: European framework shows need for safer alternatives. [Online]. Available: https://www.openaccessgovernment.org/pesticide-risk-assessment/79226/

[3] S. J. Risch, "Intercropping as cultural pest control: prospects and limitations," *Environmental Management*, vol. 7, no. 1, pp. 9–14, 1983.

[4] 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.

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

[6] M. Liebman, "Polyculture cropping systems," in *Agroecology*. CRC Press, 2018, pp. 205–218.

[7] Y. Avigal, J. Gao, W. Wong, K. Li, G. Pierroz, F. S. Deng, M. Theis, M. Presten, and K. Goldberg, "Simulating polyculture farming to tune automation policies for plant diversity and precision irrigation," in *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2020, pp. 238–245.

[8] Y. Avigal, A. Deza, W. Wong, S. Oehme, M. Presten, M. Theis, J. Chui, P. Shao, H. Huang, A. Kotani *et al.*, "Learning seed placements and automation policies for polyculture farming with companion plants."

[9] J. S. Ken Goldberg. (1995) The telegarden. [Online]. Available: https://goldberg.berkeley.edu/garden/Ars/

[10] K. Goldberg, *The Robot in the Garden: Telerobotics and Telepistemology in the Age of the Internet*. Mit Press, 2001.

[11] 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.

[12] N. Correll, N. Arechiga, A. Bolger, M. Bollini, B. Charrow, A. Clayton, F. Dominguez, K. Donahue, S. Dyar, L. Johnson *et al.*, "Building a distributed robot garden," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 1509–1516.

[13] T. Stomph, C. Dordas, A. Baranger, J. de Rijk, B. Dong, J. Evers, C. Gu, L. Li, J. Simon, E. S. Jensen *et al.*, "Designing intercrops for high yield, yield stability and efficient use of resources: Are there principles?" in *Advances in Agronomy*. Elsevier, 2020, vol. 160, no. 1, pp. 1–50.

[14] 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.

[15] 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.

[16] T. W. Ayalew, J. R. Ubbens, and I. Stavness, "Unsupervised domain adaptation for plant organ counting," *CoRR*, vol. abs/2009.01081, 2020. [Online]. Available: https://arxiv.org/abs/2009.01081

[17] M. Minervini, A. Fischbach, H. Scharr, and S. A. Tsaftaris, "Finely-grained annotated datasets for image-based plant phenotyping," *Pattern Recognition Letters*, pp. –, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865515003645

[18] M. Minervini, A. Fischbach, H. Scharr, and S. Tsaftaris, "Plant phenotyping datasets," 2015. [Online]. Available: http://www.plant-phenotyping.org/datasets

[19] H. Cuevas Velásquez, A.-J. Gallego, R. Tylecek, J. Hemming, B. Tuijl, A. Mencarelli, and R. Fisher, "Real-time stereo visual servoing for rose pruning with robotic arm," 05 2020, pp. 7050–7056.

[20] N. Strisciuglio, R. Tylecek, N. Petkov, P. Bieber, J. Hemming, E. Van Henten, T. Sattler, M. Pollefeys, T. Gevers, T. Brox, and R. Fisher, "Trimbot2020: an outdoor robot for automatic gardening," 04 2018.

[21] 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.

[22] E. Van Henten, J. Hemming, B. Tuijl, J. Kornet, J. Meuleman, J. Bontsema, and E. van Os, "An autonomous robot for harvesting cucumbers in greenhouses." *Auton. Robots*, vol. 13, pp. 241–258, 11 2002.

[23] M. Erick, S. Fiestas, R. Sixto, and G. Prado, "Modeling and simulation of kinematics and trajectory planning of a farmbot cartesian robot," *INTERCON*, vol. 1, 2018.

[24] B. Murdyantoro, D. Sukma Eka Atmaja, and H. Rachmat, "Application design of farmbot based on internet of things (iot)," *IJASEIT*, vol. 9, 2019.

[25] V. A. Murcia, J. F. Palacios, and G. BarbieriEmail author, "Farmbot simulator: Towards a virtual environment for scaled precision agriculture," *SOHOMA*, vol. 987, 2021.

[26] FarmBot. (2021) Farmbot. [Online]. Available: https://farm.bot/

[27] Niwaki. (2021) Sentei topiary clippers. [Online]. Available: https://www.niwaki.com/sentei-topiary-clippers/P00422-1

[28] FarmBot. (2021) Electronics and wiring. [Online]. Available: https://genesis.farm.bot/v1.5/Extras/bom/electronics-and-wiringcamera

[29] Pololu. (2021) Pololu carrier with sharp gp2y0a60szlf. [Online]. Available: https://www.pololu.com/product/2474

[30] Sony. (2021) Snc-vb770 ultra high sensitivity 4k network camera. [Online]. Available: https://pro.sony/enEE/products/specialised-cameras/snc-vb770

[31] ——. (2021) Fe 20mm f1.8 g full-frame large-aperture ultra-wide angle g lens. [Online]. Available: https://electronics.sony.com/imaging/lenses/all-e-mount/p/sel20f18g

[32] M. Group. (2021) Teros 10 simple soil moisture sensing. [Online]. Available: https://www.metergroup.com/environment/products/teros-10/

[33] ——. (2021) Zl6 advanced cloud data logger. [Online]. Available: https://www.metergroup.com/environment/products/zl6-data-logger/

[34] S. Mitchell, A. Rand, M. Ebeida, and C. Bajaj, "Variable radii poisson-disk sampling," *Proceedings of the 24th Canadian Conference on Computational Geometry, CCCG 2012*, vol. 13, pp. 241–258, 01 2012.

[35] T. B. Olaf Ronneberger, Philipp Fischer, "U-net: Convolutional networks for biomedical image segmentation," *arXiv preprint arXiv:1505.04597*, 2015.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[37] C. W. Whippo and R. P. Hangarter, "Phototropism: bending towards enlightenment," *The Plant Cell*, vol. 18, no. 5, pp. 1110–1119, 2006.

[38] D. Dietrich, "Hydrotropism: how roots search for water," *Journal of experimental botany*, vol. 69, no. 11, pp. 2759–2771, 2018.

[39] K. Goldberg and R. Siegwart, *Beyond Webcams: an introduction to online robots*. MIT press, 2002.

[40] J. L. Harper *et al.*, "Population biology of plants." *Population biology of plants.*, 1977.

[41] T. Kijima, *Japanese Style Companion Planting*. Tuttle Publishing, 2020.