robotics

MDPI

*Article*

# Multi-Log Grasping Using Reinforcement Learning and Virtual Visual Servoing

**Erik Wallin [1], Viktor Wiberg [1,2] and Martin Servin [1,*]**

[1] Department of Physics, Umeå University, 907 87 Umeå, Sweden; erik.wallin@umu.se (E.W.); viktor.wiberg@algoryx.se (V.W.)
[2] Algoryx Simulation AB, Kuratorvägen 2, 907 36 Umeå, Sweden
* Correspondence: martin.servin@umu.se

**Abstract:** We explore multi-log grasping using reinforcement learning and virtual visual servoing for automated forwarding in a simulated environment. Automation of forest processes is a major challenge, and many techniques regarding robot control pose different challenges due to the unstructured and harsh outdoor environment. Grasping multiple logs involves various problems of dynamics and path planning, where understanding the interaction between the grapple, logs, terrain, and obstacles requires visual information. To address these challenges, we separate image segmentation from crane control and utilise a virtual camera to provide an image stream from reconstructed 3D data. We use Cartesian control to simplify domain transfer to real-world applications. Because log piles are static, visual servoing using a 3D reconstruction of the pile and its surroundings is equivalent to using real camera data until the point of grasping. This relaxes the limits on computational resources and time for the challenge of image segmentation, and allows for data collection in situations where the log piles are not occluded. The disadvantage is the lack of information during grasping. We demonstrate that this problem is manageable and present an agent that is 95% successful in picking one or several logs from challenging piles of 2–5 logs.

**Keywords:** autonomous forwarding; visual servoing; virtual camera; reinforcement learning; multi-log grasping; Cartesian control



**Citation:** Wallin, E.; Wiberg, V.; Servin, M. Multi-Log Grasping Using Reinforcement Learning and Virtual Visual Servoing. *Robotics* **2024**, *13*, 3. https://doi.org/10.3390/robotics13010003

Academic Editors: Roman Mykhailyshyn and Ann Majewicz Fey

Received: 9 November 2023
Revised: 11 December 2023
Accepted: 12 December 2023
Published: 21 December 2023



## 1. Introduction

Automatic loading of multiple logs requires visuomotor control of a crane manipulator in a complex environment. This involves challenges in collecting and interpreting visual information for grasping and crane motion planning to handle obstacles, grapple-pile dynamics, and external conditions. While improvements in efficiency and automation are important for the forestry industry's role in sustainability, these pose major challenges due to the unstructured and harsh outdoor environment. Rough terrain with various obstacles, shaking, wear and tear of equipment, and exposure to light, weather, and seasonal conditions pose different challenges compared to the environment of conventional robot control, in particular for vision-based systems. A forwarder spends most of its time picking up logs [1], and it is crucial for high efficiency that it be able to lift multiple logs with each grasp without exceeding the maximum lift capacity of the crane. This requires detailed and unobstructed information about the piles and the environment, and makes data collection, segmentation, and crane control significant challenges that must be addressed in order to enable reliable and robust autonomous forwarding.

Driven by the global trend of big data and the progress in machine learning, the forestry industry is experiencing an increase in the collection and availability of large amounts of data. Harvest areas can be scanned from the air and the ground, and both ground and trees can be segmented [2,3], allowing detailed terrain maps to be created for path planning [4], among other things. Harvesters are increasingly being equipped with high-precision

positioning systems, and are able to store the geospatial information of the felled logs [5] as well as the travelled paths. This opens up possibilities for autonomous forwarding and increased efficiency in forestry. Removing the operator from the vehicle additionally relaxes the economic, ergonomic, and design constraints. While fully autonomous forwarding is a challenge, more imminent scenarios include operator assistance, remote-controlled machines, or partially autonomous functions.

The process of grasping logs in forestry is related to the general field of robotic grasping, which has been extensively explored in recent years [6–8]. However, there are differences that make log grasping a special case, most notably regarding grasping multiple objects, the unstructured forest environment, the electro-hydraulic crane actuation, the system size, and exposure to the elements. For the specific application of log grasping and autonomous forwarding, there are good solutions for crane motion planning and control [9,10] without considering grapple–log interaction or surrounding obstacles. Reinforcement learning (RL) control has proven to be effective for the same task in simulations, grasping a single log with known pose [11]. However, transferring such joint-level RL control to a real system is a problem due to simulation bias when the electro-hydraulic circuit [12,13] has not been precisely modelled. Dhakate et al. [14] shows how joints can be modelled and the dynamics learned using RL to enable Cartesian control. Actuator dynamics are specific to each machine, non-intuitive for humans, and difficult to interface with other control systems or human operators for shared control of crane operation [15]. Cartesian control, on the other hand, can be seen as a common interface, which is more intuitive and interfaces more easily with other systems. Considering the grapple, logs, and obstacles, there is a need for visual input to take their configurations and interactions into account. Logs may be partially overlapped or interlocked, and successful grasps may depend on small geometric details that affect the interaction between the grapple and the logs. At the same time, the terrain and obstacles, such as trees and rocks, make the grasping task more than a grasp-pose estimation problem, additionally involving a crane control problem with grasp dynamics and path planning. While there are methods for log detection [16,17], varying conditions and occlusion make real-time segmentation difficult and hinder continuous crane and grasping control. There are, however, promising experiments in which segmentation has been used to identify grasp poses. La Hera et al. [18] shows sparks of early autonomous forwarding in practice, picking single logs along a path on flat ground in concept machine experiments. Ayoub et al. [19] developed a grasp planning algorithm which was successfully tested on a physical crane to grasp single or multiple logs on flat ground. In this approach, logs are segmented and modelled in a simulator to produce depth-camera images, from which a grasp pose is generated by a convolutional neural network (CNN).

Visual information for continuous crane and grasping control should provide a good overview and be unobstructed, including occlusion by the crane and grapple. It would be beneficial to collect visual data during moments with good visibility or to combine data from different times and perspectives. Another option would be to separate segmentation and control, using specialised systems for each. Considering this, we define a *virtual camera* as a sensor that generates a stream of 2D data originating from a 3D reconstruction; see Figure 1.

To address the challenge of collecting and using visual data for control in challenging forest environments, we explore using reinforcement learning and virtual visual servoing for multi-log grasping. We utilise Cartesian control to simplify the typical reinforcement learning problems of simulation-to-reality (sim-to-real) transfer and interfacing with other control systems or human operators. To address the issue of occlusion in visual servoing for crane control, we utilise a virtual camera, allowing the underlying 3D reconstruction data to be captured where there is no obstruction. This enables data from different times or perspectives to be combined and removes the need for real-time segmentation, allowing more time and computational resources for this task. We train agents using multibody dynamics with frictional contacts, with a reward signal designed to provide dense feedback

from the camera data. In addition, we investigate ways to gain insight into learned behaviours, with a focus on the use of image data.

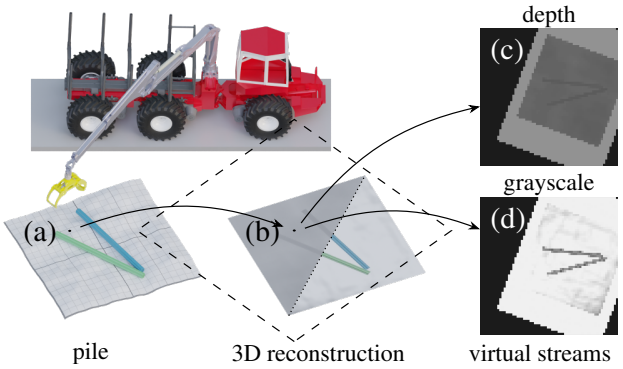


**Figure 1.** Illustration of the virtual camera setup, showing (**a**) the actual pile, (**b**) the corresponding 3D reconstruction, and (**c**,**d**) the depth and greyscale virtual streams. The position of the virtual camera is represented by a dot, with the orientation and extent illustrated by the dashed square.

## 2. Method

To test control from 3D reconstructed data using virtual cameras, we train an agent to grasp multiple logs using model-free RL. Application in practice would require segmenting logs and removing disturbing background from real image data [16,17]. Here, we work with piles generated to match such corresponding output. We generate log piles and simulate a forwarder using multibody dynamics with frictional contacts using the AGX Dynamics physics engine [20].

### *2.1. Piles and Virtual Camera*

We used Perlin noise [21] to generate uneven terrain as $5 \times 5$ m$^2$ patches, and formed disordered piles with 2–5 logs by stacking logs vertically with random displacements and rotations in the horizontal plane, then letting them fall to the ground. To emulate output from log segmentation, the ground was coloured in a uniform bright colour, then colour and depth (RGB-D) images were generated using an orthographic camera placed straight above the pile, as seen in Figure 2. The displacement components and rotation for the logs were sampled from Gaussian distributions centred around zero with $\sigma_{\text{pos}} = 0.5$ m and $\sigma_{\text{rot}} = 0.25$ rad, determined empirically to achieve varying and challenging piles. To make logs less prone to rolling, they were modelled by two overlapping square cuboids with a relative rotation of 45°. We delimit ourselves to fixed-sized and shaped logs, using cuboids that are 3.5 m long and $\sqrt{2}/10$ m thick to emulate logs with a diameter of 0.2 m and a mass of 112 kg. Cases where the logs did not relax quickly were discarded by comparing the mean log speed to a small threshold $\epsilon_v = 5 \times 10^{-3}$ m/s within 10 s. The target grasp pose was set according to the position and orientation of the log closest to the combined log centre of mass position which was not occluded by any other log; see Figure 2.

The aim of the virtual camera was to imitate the output of a real camera as if mounted on the grapple while using segmented 3D reconstructed data; see Figure 1. The relative position $\mathbf{r}_{\text{rel}}$ and orientation $\phi_{\text{rel}}$ of the pile and the virtual camera were used to transform the RGB-D data to a virtual camera output stream. To reduce the dimensionality of the camera data, the RGB data were converted to greyscale. The RGB colours of the logs were sampled from small ($\sigma = 10\%$) Gaussian variations around grey. This ensured that all logs were similar in greyscale, emphasising that logs must not be individually segmented.

The orthographic camera lacks perspective, and is simply specified by its resolution and physical size. We set the resolution to $64 \times 64$ pixels; to mimic a field-of-view, we varied the camera size depending on the z-component of $\mathbf{r}_{\text{rel}}$. This was done by defining the camera sizes $s_{\text{far}}$ and $s_{\text{near}}$ at some *far* (5 m) and *near* (0 m) distances and using linear interpolation in between. A virtual camera is not limited to obeying the constraints of

physics, as a real camera is. This flexibility allows for the exploration of scenarios that may be challenging or unattainable to replicate in the physical world. We explored $s_{\text{far}} = 15$ m and $s_{\text{near}} = 3$ m in order to retain an overview during the grasp moment, when the grapple is close to the pile. While the RGB sensor data was independent of the distance to the pile, the depth sensor data were rescaled to match the output of a real depth camera. A major difference between a virtual camera and a physical one is that the underlying data of the virtual camera do not update after grasping. We investigated how important the different observables were to the agent's behaviour at different stages of the grasping cycle.

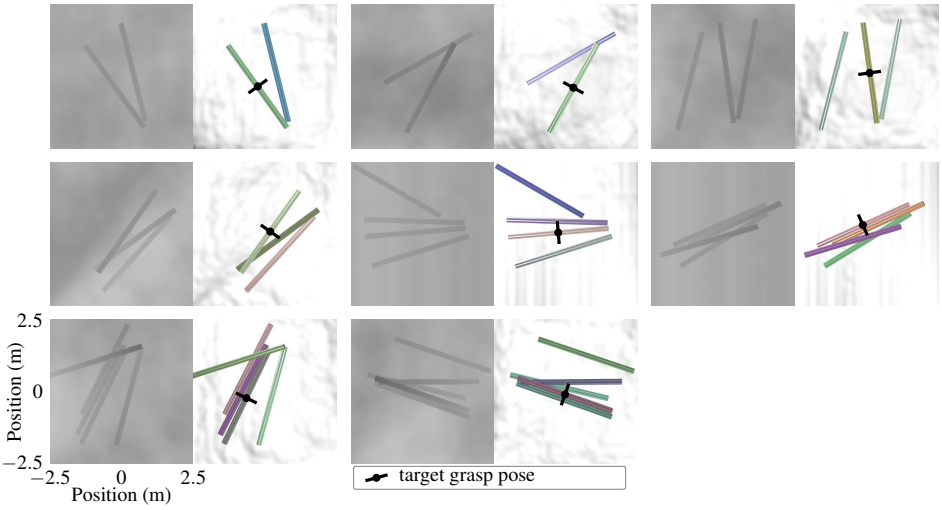


**Figure 2.** Example of piles, with corresponding depth and RGB images for eight piles with 2–5 logs. The elevation difference of the used terrains ranges from 0.2 m to 0.8 m, with a mean of 0.4 m.

### *2.2. Crane Control and Calibration*

The crane is a *Cranab FC12* (Cranab AB) mounted on an *Xt28* (eXtractor AB) pendulum arms concept forwarder; see Figure 3. It consists of 21 bodies and 26 joints, of which 6 are actuated. The *pillar* is connected to the *base*, and can rotate by an actuated hinge (a). From the pillar, the *main boom* is connected with a hinge (b) and a piston that provides hydraulic power. The *outer boom* similarly connects (c) from the main boom, and the *telescope* can extend (d) from the outer boom, powered by a piston. The end-effector consists of a *rotator* and a *grapple*. The rotator has one actuated hinge (e) for rotating the grapple and two hinges (g–h) that allow the grapple to swing. The grapple opens and closes (f), powered by a piston. To speed up simulations, the mesh geometry of the grapple was replaced by a similar simplified geometry made up of nine boxes, while the original geometry was retained for visuals.


**Figure 3.** The *Xt28* concept forwarder with the *Cranab FC12* crane mounted. The semi-transparent blue boxes show the simplified grapple geometry. The letters represent actuated joints (**a**–**f**) and passive joints (**g**,**h**).

Joint range and force limits were calibrated using data from the manufacturer [22], though these were not experimentally confirmed. Joint range limits were set using the maximum reach of the crane and illustrations/images of different configurations, while force limits were set guided by data of the lift capacity at some discrete crane configurations. The lowest lift capacity, at the 8 m full extension, was 9.7 kN. As the logs weigh 112 kg, this lift capacity is enough to easily lift five logs even at full extension. To model the friction in the rotator hinges, we used weak lock constraints and tuned the force limits and compliance until the damping of the swinging of the grapple appeared physical and agreeable with video material. The crane weighs 1630 kg, while the rotator and grapple weigh 249 kg together.

We implemented *Cartesian control*; thus, from a desired crane-tip velocity $\mathbf{v}_{\text{crane-tip}}$ in Cartesian world frame coordinates, the corresponding target velocity of each joint is calculated with inverse kinematics [23]. As an alternative to joint-level control, Cartesian control is becoming increasingly common in commercial forest machines [24]. Actuator dynamics are specific to each machine design, whereas Cartesian control can be seen as a layer of abstraction, exposing a common interface. This increases generality and simplifies implementation and sim-to-real transfer, removing the need for precise modelling of the electro-hydraulic crane actuation. In addition, it simplifies combining control with human operators or other control systems [15], e.g., for obstacle avoidance.

The Cartesian control problem for the described crane, with four degrees of freedom to control the three components of the crane-tip velocity, is an *under-determined* system. Thus, there is no inverse to the Jacobian describing how the crane-tip velocity is affected by the velocity of each joint given some crane configuration, i.e., there can be (infinitely) many joint velocity solutions for a single crane-tip velocity. This issue was addressed by defining a *pseudo-inverse*, with weights for prioritising motion in different joints. We defined these as functions of the articulation of each joint, which are approximately constant but decrease to 10% near the range limits. This makes the system solvable, with solutions mostly within the physical limits of the actuators.

To simplify the modelling and avoid slowing down simulations, we modelled the crane hydraulics using kinematic constraints instead of hydraulic and electric circuit simulations. For each actuator, the force/torque was determined as a solution of the multibody dynamics equation while considering the provided limits on joint ranges and motor force. To mimic the relatively slow motion of the hydraulics, the requested joint velocities were restricted by clipping in the range of $[-1, 1]$ m/s (rad/s).

### *2.3. Reinforcement Learning Control*

Reinforcement learning is a machine learning method in which an agent learns through trial and error. It has proven successful in complex control problems with high-dimensional observations such as visual data where otherwise conventional control systems have struggled. The agent selects an *action* based on a *state* and its *observation* of it. A *reward* signal is used to guide the learning towards desired state–action mappings [25]. RL has led to many impressive results, especially in games [26], though it has yet to be widely used in real-world applications. Compared to classical control methods, its main strengths are in complex planning tasks with long horizons and many degrees of freedom.

#### 2.3.1. Observation and Action

The observation space consists of the virtual camera output and sixteen scalar values concerning the crane, grapple, and target configurations. The camera data are $64 \times 64$ pixels with two channels. To maintain the idea of Cartesian control as a high-level interface, we chose not to include joint observations of the crane, i.e., the angles/speed of the joints (a–d) in Figure 3. Instead, we used the grapple's relative position, velocity, and speed with respect to the target. Details regarding the rotator and grapple are provided, along with the angles and angular speed for the rotation, swing (two directions), and grapple opening. Furthermore, to compensate for the lack of joint observations and not deprive the agent of

all haptic sense, we provide a virtual load cell in the rotator. This measures the grapple–load weight, which is normalised by subtracting and dividing by the empty grapple weight. In practice, the crane configuration and the pressure in the hydraulic cylinder of the main boom could provide such force estimates. Angle and speed observations for the grapple and rotator joints were scaled to $[-1, 1]$ using their respective limits, while other observations were clipped to $[-10, 10]$ to encompass the full range of the typical relative grapple position components. The relative rotation of the grapple to the target angle was *not* included as one of the observations. The motivation behind this was to create a dense dependence on the camera data containing information on the angles of all logs compared to the grapple. We suggest that this increases the ability of the agent to analyse the camera data, which simplifies the learning process.

The action consists of five scalar values, where three represent the velocity components of the desired crane-tip velocity and the other two represent rotating and opening/closing the grapple.

#### 2.3.2. Reward

We designed a reward function

$$r = r_{\text{target}} + r_{\text{guide}} + r_{\text{energy}} \tag{1}$$

that combines a sparse term related to overall success or failure with dense terms to aid learning from image data. The sparse term $r_{\text{target}}$ is designed to become the dominant term, with the others intended to aid learning without overly biasing the final behaviour. The relative contributions to the accumulated reward depend on the learned behaviour, and cannot be immediately inferred. For the trained agent, they are 92%, 10%, and −2%, respectively.

We used zero-centred Gaussian functions for scaling, denoting these as $G(x; \sigma) = e^{-0.5(x/\sigma)^2}$ for some measure $x$, or $G_\sigma$ for short. The first term, $r_{\text{target}}$, is awarded only when the agent has achieved the target objective of grasping one or several logs and lifting them a sufficient height off the ground:

$$r_{\text{target}} = 25G(x_{\Delta\text{grasp}}; \sigma_{\Delta\text{grasp}}) + 1.12N_{\text{logs}} \tag{2}$$

where $x_{\Delta\text{grasp}}$ is the proximity of the grapple to the centre of mass of the logs in the grapple, $\sigma_{\Delta\text{grasp}} = 0.5$ m, and $N_{\text{logs}}$ is the number of logs in the grapple.

The second term in Equation (2), $r_{\text{guide}}$, is a dense reward designed to help the agent consistently learn to grasp logs:

$$r_{\text{guide}} = r_{\text{stage}} G_{\Delta\text{tilt}} / N_{\text{steps}} \tag{3}$$

where $G_{\Delta\text{tilt}}$ scales with the vertical tilt of the grapple, $\sigma_{\Delta\text{tilt}} = 0.2$, $N_{\text{steps}}$ is the number action steps, and $r_{\text{stage}}$ is any of three stages. Stage 1 provides an increasing reward for proximity to the target position, aligning with the target angle, and opening the grapple; Stage 2 provides an increasing reward for closing the grapple; and Stage 3 is activated when the grapple has closed around at least one log, with an increasing reward for lifting the grapple. We believe that the use of a dense reward term is vital for learning appropriate grapple angles from image data, where the dense reward greatly increases the feedback as to which grapple angle the image data represents. The third term in Equation (2) is a penalty for excessive energy use, which is proportional to the sum of the power of the actuators.

#### 2.3.3. Curriculum

Each episode of the RL task features a pile placed according to a function, with a *difficulty parameter* $d \in [0, 1]$ determining the challenge level. To speed up the simulations, we kept the vehicle in the same configuration and placed the pile in relation to it. For $d = 0$,

the pile was always placed just below the starting position of the grapple, while for $d = 1$ it was placed with random rotation at challenging positions on either side of the vehicle at varying heights $z \in [-1/2, 1]$ m. For intermediate difficulty levels, a linear interpolation of the two cases was used, allowing the challenge of the task to be smoothly adjusted. Collisions between the vehicle and the crane/piles were disabled, as piles can overlap with the vehicle, especially during the curriculum.

The curriculum consisted of lessons, during which we adjusted the difficulty parameter in increments of 0.1. Twenty evaluation episodes were conducted every 50,000 steps, and progress to the next lesson was determined by the mean accumulated reward of the past $10 \times 20$ evaluation episodes compared to a threshold. The threshold was empirically determined and set to 21 to allow progress through the curriculum on a regular basis. In addition to varying the target position, we modified the criterion for target success. As the lessons became more challenging, we required the logs to be raised higher above the ground, from 0.25 m for $d = 0$ to 1.1 m for $d = 1$.

#### 2.3.4. RL Algorithm and Network

We used the Stable-Baselines3 [27] RL library with the *model-free on-policy* algorithm PPO [28]. While this setup can enable learning in complex environments, it tends to be sample-inefficient. Unlike model-based methods, it does not build an internal model of the environment, instead learning a mapping from states to actions in order to maximise the expected accumulated discounted reward. After each policy update in PPO, new data must be acquired using the new policy.

The input data for our RL agent consisted of sixteen floating-point numbers and two channels of $64 \times 64$ images. The images pass through a CNN *feature extractor* network, and the resulting vector is concatenated with the other observations. The concatenated input is then fed into two fully connected neural networks, one to predict the value function and the other to generate the action.

We carried out training using eight environments with a maximum episode length of 10 s, a simulation frequency of 60 Hz, and a control frequency of 20 Hz. A number of hyperparameters, such as the batch size, learning rate, and network parameters, were varied to find the agents with the best performance. The best model was trained using a batch size of 1600, a learning rate of 0.00025, and a feature extractor CNN with (8, 8, 8) filters of sizes [8, 4, 3] and strides [4, 2, 1], and 64 output features. The fully connected networks have two hidden layers of size (64, 64), with tanh activation functions. A summary of the hyperparameters can be found in Table 1.

**Table 1.** Hyperparameters; for details, see [29].

| Hyperparameter | Value | Hyperparameter | Value |
|---|---|---|---|
| n-envs | 8 | episode-length | 200 |
| batch-size | 1600 | learning-rate | 0.00025 |
| gamma | 0.99 | n-epochs | 4 |
| ent-coef | 0.0 | vf-coef | 0.5 |
| max-grad-norm | 0.5 | gae-lambda | 0.95 |
| clip-range | 0.2 | | |

## 3. Results and Discussion

In this section, we present the results from training and evaluating an agent. We analyse the importance of the observations and try to shed light on the inner workings of the agent.

### *3.1. Training*

We trained agents for up to 20 million steps, and extended the training of successful agents. Figure 4 shows training curves for the selected agent, which achieved the highest smoothed reward. The agent was first trained on piles with two logs placed at a restricted

radius range $r \in [4.5, 5.5]$. Loading from the best-performing stage, training was resumed with piles of 2–5 logs and the full radius range. After again reloading from the best-performing stage and passing the curriculum, the best agent was selected. Having passed through all the curriculum lessons, the agent reward and success rate fluctuated at a high level, though without achieving consistent mastery.


**Figure 4.** Evaluation curves during training of the selected agent, showing the reward, lesson number, and smoothed reward using a sliding window of size 10: (**a**) shows training with two logs and restricted radius range, while (**b**,**c**) show training with 2–5 logs. The grey regions highlight the final lesson with a non-simplified task. The lesson number maps to the difficulty parameter $d$, as described in Section 2.3.3.

### *3.2. Evaluation*

To evaluate the agent's performance, we conducted 1000 grasp attempts on evaluation piles with 2–5 logs. The test setup was similar to the training setup, with a different set of piles that were not used during training. Success was defined as the agent grasping one or more logs and lifting them to an elevation gain of 1.1 m. The overall success rate was 95 %, as shown in Figure 5, and the most common yield was two grasped logs. The success rate was the highest (97%) for piles of three logs and the lowest (91%) for piles of five. Figure 6e shows the target grasp position for each attempt, coloured based on the accumulated reward and with failed attempts shown as $\times$. The agent learned to pick logs over the entire area, with no apparent systematic pattern to the failures. The design of the target distribution function sometimes results in logs close to/underneath the vehicle. If collisions between the crane and the vehicle are enabled, there can be collisions for targets within 1.25 m of the wheels. For low targets in the very back, the main boom can collide with the load bunk due to under-use of the telescope in the IK implementation.

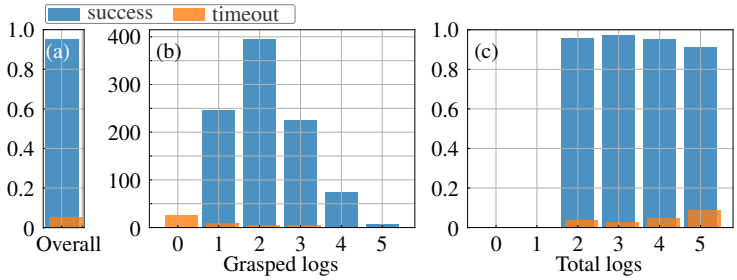


**Figure 5.** (**a**) Overall success of 95%; (**b**) number of logs grasped; and (**c**) success relative to the number of logs in the pile.

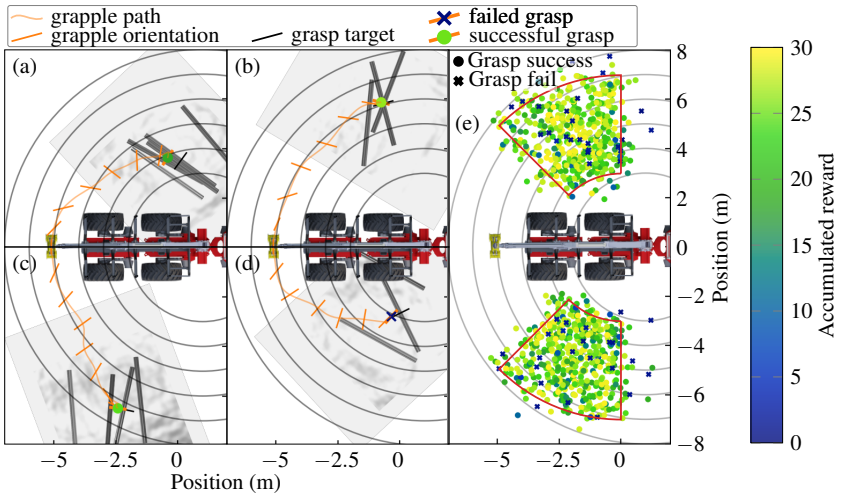


**Figure 6.** (**a**–**d**) Example of four grasp attempts and (**e**) illustration of target locations. The grapple path and orientations are shown in yellow, with suggested/actual grasp poses in black/thick yellow. Target locations and grasps are coloured after accumulated reward, with failures illustrated by ×. The red outline marks the region where piles were placed, with target locations outside of this due to offsets from pile centres.

As the grasp pose was set according to the position and orientation of one of the logs, this is most likely not the optimal pose for grasping multiple logs. Thus, in order to increase the probability of picking multiple logs, the agent must learn to make deviations from the suggested grasp pose using the camera data. Figure 6 and the Supplementary Materials show details of four specific attempts, three successful and one failed. It is important to be careful when drawing conclusions from individual evaluations, as the agent and its interaction with the environment are complicated and can give rise to seemingly random behaviour. Nonetheless, Figure 6c seems to show a small deviation to better grasp two logs instead of one, and Figure 6b seems to show the grapple rotation being adjusted to better grasp both logs. To determine whether this is a coincidence or a learned strategy, we introduced systematic perturbations in the target grasp position and studied the resulting spatial distribution of the actual grasp positions for the specific case of Figure 6c. The resulting grasp positions were mainly drawn towards a position in between the leftmost logs or on the log to the right, as can be seen from the heatmap in Figure 7. From this, it can be concluded that the agent is able to utilise the camera data to make strategic deviations from a given target position. Typical variations in grasp position compared to the target position for the 1000 evaluation grasps were in the range ±0.5 m. Thus, the agent's sensitivity to the recommended grasp position is limited. The grasp sequence in a 3D view corresponding to the case in Figure 6c can be seen in Figure 8.

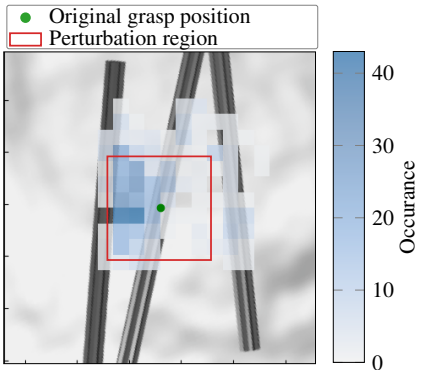


**Figure 7.** Heatmap showing the grasp position of the agent for 625 grasp attempts where the original target position was systematically perturbed within a 1 × 1 m region for the same pile as in Figure 6c.

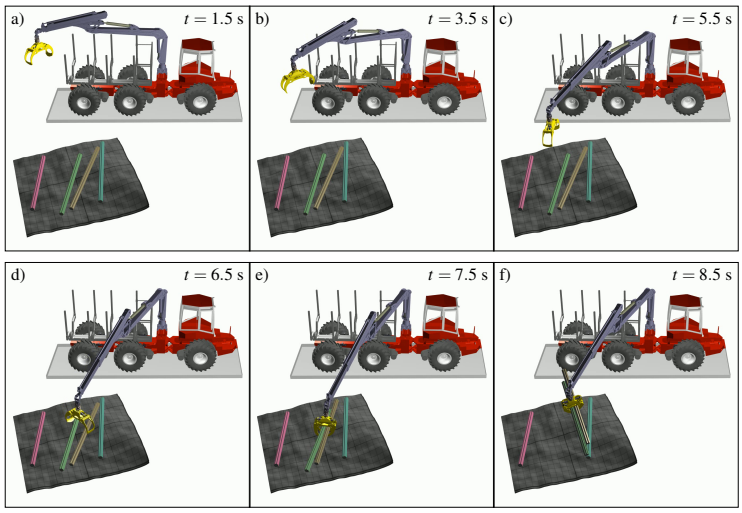


**Figure 8.** (**a**–**f**) Image sequence of one example grasp for the same pile as in Figure 6c.

### *3.3. Observation Ablation Study*

To understand the importance of different observations for the agent, we conducted three types of ablation experiments. First, we trained the agent with and without particular observations in order to measure their importance. Second, we added noise to observations during evaluation and measured the resulting loss in performance. Finally, we added noise to observations in already recorded data and measured the resulting change in actions.

Retraining the agent without certain observations or with additional observations is computationally expensive and time-consuming. Therefore, we only performed these experiments selectively to verify specific design decisions made in setting up the agent, such as not providing the target angle as an observation to impose a greater dependency on the camera data. We additionally used it to verify the agent's use of camera data. There is a bias in favour of the baseline, in the sense that the hyperparameters and the curriculum were set up to achieve success for the baseline, whereas this may not be optimal for other cases. Nonetheless, where there are significant differences this can provide a decent indication of the importance of adding or removing an observation in this particular setup. The results were measured based on the total number of lessons passed, including repeatedly passing the final lesson, and are presented in Table 2. It can be seen that adding the target angle as an observation is detrimental to learning. This aligns with our idea that a dense dependence on camera data is crucial for learning to use it effectively. Removing the depth camera seemed to create more challenging conditions than removing the greyscale camera, indicating the importance of the depth camera in this stage of training. Removing both cameras and instead relying on the target angle results in very poor training. It is definitely possible to learn such a task, as shown in Andersson et al. [11] for a single log on flat ground, and the bad performance seen here could be an example of the bias towards the baseline, as mentioned earlier. However, it verifies the use and importance of the camera data in this particular setup, and demonstrates how the baseline agent learns features in the camera data that are not captured by the target angle alone.

To gain further insight into the importance of different observations, we added different levels of noise to the observation signals during our evaluations. The idea behind this approach is that the reward becomes sensitive to noise for important observations and insensitive to noise for less important or redundant observations. While there is no impartial level of noise that would enable a perfect comparison between different observations, we tried to find fair levels based on the distribution of each observable's values. We found the standard deviation $\sigma_i$ of each observation $o_i$ through an evaluation using 1000 grasp attempts and used this to scale the noise for each observation. The added noise was drawn from a Gaussian distribution, and we considered noise at eight different levels in the range $[2^{-4}, 2^3] \times \sigma_i$. To determine the performance for a given observation and level of noise,

we performed 100 evaluations and measured the mean reward. The results can be seen in Figure 9. The relative position is clearly very important. Other important observations are the grapple–load weight and the opening angle of the grapple. In contrast, observations related to the swing angle, swing speed, and rotation of the grapple seem not to be as important to the agent. The latter shows how the grapple rotating action does not depend on the rotation angle, but rather the camera data and the rotation speed.

**Table 2.** Results of adding (+) or removing (−) observations on the total amount of lessons passed during 20 M steps of training. The trainings were repeated five times, and the mean and standard deviation are displayed.

| # | Case | Lesson Success (std) |
|---|---|---|
| 0 | baseline | 91.2 (39.7) |
| 1 | + target angle | 18.6 (10.2) |
| 2 | + joint angles | 10.4 (2.0) |
| 3 | − depth camera | 10.4 (6.0) |
| 4 | − greyscale camera | 47.0 (30.9) |
| 5 | − cameras, +target angle | 1.8 (3.6) |

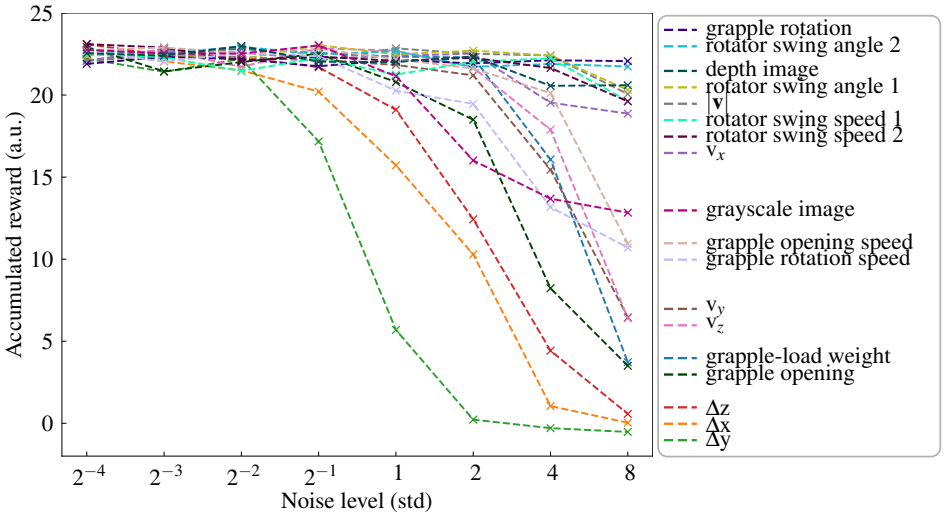


**Figure 9.** Mean accumulated reward over 100 evaluations while adding different levels of noise to each observable in turn.

Finally, we used recorded data from 1000 evaluations and added noise to each observation in turn in order to observe how the actions of the agent changed. In addition to highlighting important observations, this can reveal when during the load cycle an observation is most important and for what actions, providing insights into the inner workings of the agent. The noise was drawn from a Gaussian distribution with $\mu = 0$ and $\sigma_i = 0.2(\max(o_i) - \min(o_i))$. As can be seen from the results in Figure 10, the importance of the relative position and the grapple-load weight is again highlighted. The relative position is obviously important for the crane-tip actions, and there is a clear importance for the open–close grapple action as well. Adding noise to the greyscale image channel has a larger effect on the actions than adding noise to the depth camera, which is consistent with the corresponding larger drop in reward seen in Figure 9. However, the results in Table 2 do suggest that the depth camera is more important for passing the curriculum during training. From Figure 10, it can be seen that the greyscale camera is important for positioning and aligning the grapple with the logs as well as for timing the closing of the grapple, while the depth camera is mostly important for timing the closing of the grapple. This is not unreasonable, considering that the greyscale camera shows greater contrast between the logs and the background, while the depth camera provides information about

the distance to the logs. It might be the case that the depth camera is more important for passing the curriculum, even if the agent has greater dependence on the greyscale camera during the latter stage of training.

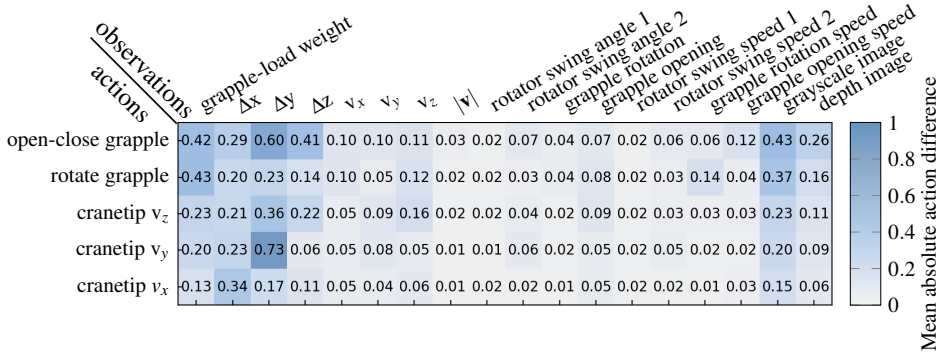


**Figure 10.** Mean absolute difference in action when adding noise to each observation in turn on recorded data from 1000 evaluations.

## 4. Conclusions

We conclude that using a virtual camera stream from 3D reconstructed data is a viable setup for multi-log grasping, with the agent able to use the camera data for grasping despite the underlying data not updating during the grasp as a real camera would. The agent learns to pick logs with 95% accuracy, using the camera when steering the crane tip as well as when rotating and closing the grapple. The Cartesian control simplifies domain adaption for deploying the RL agent on a real machine. Using a virtual camera allows for collecting visual information when the view is not occluded, combining data from different times or perspectives, and working with processed data to avoid real-time segmentation. This enables solutions to problems related to segmentation, occlusion, season, weather, and light conditions in applications in unstructured forest environments.

The grasping agent has a modular design that is interoperable with any method for crane control that takes the crane-tip target velocity as input. This includes existing methods for time-optimal trajectory planning and control [9] and semi-autonomous shared control [15], with the possibility of introducing geofences around the machine and other known objects. This interoperability is important to ensure the safety and productivity of the automated system, e.g., through human monitoring of planned motion with the possibility of intervening by manually adjusting the speed and direction of the crane-tip motion. The implication is that automatic loading can be introduced as an assistive system well before the system is sufficiently mature for autonomous control.

Our observation ablation/augmentation study provides insights into the inner workings of the agent, showing how a dense dependence on camera data is important for allowing the agent to utilise vision and how the agent uses features of the camera data that are not captured by the target angle alone. Our observation noise study reveals the importance of each observation, indicating that the grapple–load weight is a vital observation and that the greyscale camera is more important for the trained agent than the depth camera. Additionally, the study results show that the grapple rotating action is controlled by the camera data and rotation speed, and does not involve the rotation angle itself.

Possible future work involves improvements in RL methods and training to achieve master-level performance, the inclusion of models for optimal grasp poses, the inclusion of log diversity in terms of size and shape, and transfer of the learned skills to a real machine. Transfer tests of the learned skills to a real machine will involve integration with a log segmentation algorithm such as the one described in [17] and interfacing with a crane control system that takes the crane-tip velocity as an input. In addition to RGB-D sensing, the test system will need to be equipped with sensors for the grapple's orientation and opening as well as an estimator for the load weight.

**Supplementary Materials:** The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/robotics13010003/s1, Video S1: Supplementary Video.

**Author Contributions:** Conceptualization, M.S.; Methodology, E.W., V.W. and M.S.; Software, E.W. and V.W.; Formal Analysis, E.W.; Writing—Original Draft Preparation, E.W.; Writing—Review and Editing, E.W., V.W. and M.S.; Visualization, E.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by Mistra Digital Forest, Grant DIA 2017/14 #6. The simulations were performed on resources provided by the Swedish National Infrastructure for Computing at High Performance Computing Center North (HPC2N).

**Data Availability Statement:** The data presented in this study may be made available on request from the corresponding author. The data are not publicly available due to the need for additional time and resources to clean and refactor the code.

**Acknowledgments:** This work was supported by Algoryx Simulation AB, Cranab AB, and eXtractor AB.

**Conflicts of Interest:** Authors Viktor Wiberg and Martin Servin were employed by the company Algoryx Simulation AB. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## References

1. Lundbäck, M.; Häggström, C.; Fjeld, D.; Lindroos, O.; Nordfjell, T. The economic potential of semi-automated tele-extraction of roundwood in Sweden. *Int. J. For. Eng.* **2022**, *33*, 271–288. [CrossRef]
2. Axelsson, P. Processing of laser scanner data—Algorithms and applications. *ISPRS J. Photogramm. Remote Sens.* **1999**, *54*, 138–147. [CrossRef]
3. Elmqvist, M.; Jungert, E.; Lantz, F.; Persson, A.; Soderman, U. Terrain modelling and analysis using laser scanner data. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2001**, *34*, 219–226.
4. Wallin, E.; Wiberg, V.; Vesterlund, F.; Holmgren, J.; Persson, H.J.; Servin, M. Learning multiobjective rough terrain traversability. *J. Terramech.* **2022**, *102*, 17–26. [CrossRef]
5. Lindroos, O.; Mendoza-Trejo, O.; La Hera, P.; Morales, D.O. Advances in using robots in forestry operations. In *Robotics and Automation for Improving Agriculture*; Burleigh Dodds Science Publishing: Cambridge, UK, 2019; pp. 233–260.
6. Caldera, S.; Rassau, A.; Chai, D. Review of deep learning methods in robotic grasp detection. *Multimodal Technol. Interact.* **2018**, *2*, 57. [CrossRef]
7. Levine, S.; Pastor, P.; Krizhevsky, A.; Ibarz, J.; Quillen, D. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *Int. J. Robot. Res.* **2018**, *37*, 421–436. [CrossRef]
8. Kleeberger, K.; Bormann, R.; Kraus, W.; Huber, M.F. A survey on learning-based robotic grasping. *Curr. Robot. Rep.* **2020**, *1*, 239–249. [CrossRef]
9. Ortiz Morales, D.; Westerberg, S.; La Hera, P.X.; Mettin, U.; Freidovich, L.; Shiriaev, A.S. Increasing the level of automation in the forestry logging process with crane trajectory planning and control. *J. Field Robot.* **2014**, *31*, 343–363. [CrossRef]
10. Taheri, A.; Gustafsson, P.; Rösth, M.; Ghabcheloo, R.; Pajarinen, J. Nonlinear Model Learning for Compensation and Feedforward Control of Real-World Hydraulic Actuators Using Gaussian Processes. *IEEE Robot. Autom. Lett.* **2022**, *7*, 9525–9532. [CrossRef]
11. Andersson, J.; Bodin, K.; Lindmark, D.; Servin, M.; Wallin, E. Reinforcement learning control of a forestry crane manipulator. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 2121–2126.
12. Zhao, W.; Queralta, J.P.; Westerlund, T. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. In Proceedings of the 2020 IEEE Symposium Series on Computational Intelligence (SSCI), Canberra, ACT, Australia, 1–4 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 737–744.
13. Wiberg, V.; Wallin, E.; Fälldin, A.; Semberg, T.; Rossander, M.; Wadbro, E.; Servin, M. Sim-to-real transfer of active suspension control using deep reinforcement learning. *arXiv* **2023**, arXiv:2306.11171.
14. Dhakate, R.; Brommer, C.; Bohm, C.; Gietler, H.; Weiss, S.; Steinbrener, J. Autonomous Control of Redundant Hydraulic Manipulator Using Reinforcement Learning with Action Feedback. In Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022; IEEE: Piscataway, NJ, USA 2022; pp. 7036–7043.
15. Hansson, A.; Servin, M. Semi-autonomous shared control of large-scale manipulator arms. *Control Eng. Pract.* **2010**, *18*, 1069–1076. [CrossRef]
16. Ainetter, S.; Böhm, C.; Dhakate, R.; Weiss, S.; Fraundorfer, F. Depth-aware object segmentation and grasp detection for robotic picking tasks. *arXiv* **2021**, arXiv:2111.11114.
17. Fortin, J.M.; Gamache, O.; Grondin, V.; Pomerleau, F.; Giguère, P. Instance segmentation for autonomous log grasping in forestry operations. In Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 6064–6071.

18. La Hera, P.; Trejo, O.; Lindroos, O.; Lideskog, H.; Lindbäck, T.; Latif, S.; Li, S.; Karlberg, M. Exploring the Feasibility of Autonomous Forestry Operations: Results from the First Experimental Unmanned Machine. *Authorea* **2023**. [CrossRef]
19. Ayoub, E.; Levesque, P.; Sharf, I. Grasp Planning with CNN for Log-loading Forestry Machine. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 11802–11808.
20. Algoryx Simulations. AGX Dynamics. Available online: https://www.algoryx.se/agx-dynamics/ (accessed on 30 October 2023).
21. Perlin, K. An image synthesizer. *Acm Siggraph Comput. Graph.* **1985**, *19*, 287–296. [CrossRef]
22. Cranab AB. Forwarder Cranes Brochure. Available online: https://www.cranab.com/downloads/Forwarder-Cranes/Cranab-FC-brochure-EN.pdf (accessed on 8 October 2022).
23. Spong, M.W.; Vidyasagar, M. *Robot Dynamics and Control*; John Wiley & Sons: Hoboken, NJ, USA, 2008.
24. Palmroth, M.; Laitinen, S.; Siltanen, V.; Käppi, T. Method and System for Controlling the Crane of a Working Machine by Using Boom Tip Control. Patent WO2014118430A1. 7 August 2014. Available online: https://patents.google.com/patent/WO2014118430A1/ko (accessed on 30 October 2023).
25. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
26. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef]
27. Raffin, A.; Hill, A.; Gleave, A.; Kanervisto, A.; Ernestus, M.; Dormann, N. Stable-Baselines3: Reliable Reinforcement Learning Implementations. *J. Mach. Learn. Res.* **2021**, *22*, 1–8.
28. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.
29. Stable baselines3. PPO Documentation. Available online: https://stable-baselines3.readthedocs.io/en/master/modules/ppo.html (accessed on 30 October 2023).