Cornhole: A Widely-Accessible AI Robotics Task

Nate Derbinsky
Wentworth Institute of Technology
550 Huntington Ave
Boston, MA 02115

Tyler M. Frasca
Tufts University
419 Boston Avenue
Medford, MA 02155

Abstract

In this paper we present the game of cornhole as a compelling, accessible, and adaptable AI robotics task. Cornhole is a fun and social game with simple rules, but involves strategy and physical training for humans to play competitively; thus, developing a robot that can play at the level of even the average human player presents a multitude of opportunities for curricular integration at a variety of levels. We characterize the AI tasks involved with the game, and present results and resources gained from preliminary offerings.

Cornhole[1] is a game in which players take turns throwing bean bags at angled platforms, each with a hole (e.g. see Figure 1). Though typically a casual social game, competitive cornhole has also been gaining traction – for example, ESPN covered the American Cornhole Organization's World Championships of Cornhole (2015). Despite this attention, to our knowledge, no one has attempted to build a robot that can play the game, and so we see cornhole as an exciting, yet accessible, AI robotics task for undergraduate education with a diverse and adaptable set of challenges.

Figure 1: Example play, one player "pitches" a bag at a time.

[1]Depending upon the region, the sport may be known by a variety of names, including dummy boards, [bean] bag toss, dadhole, doghouse, Baggo, or Bags.

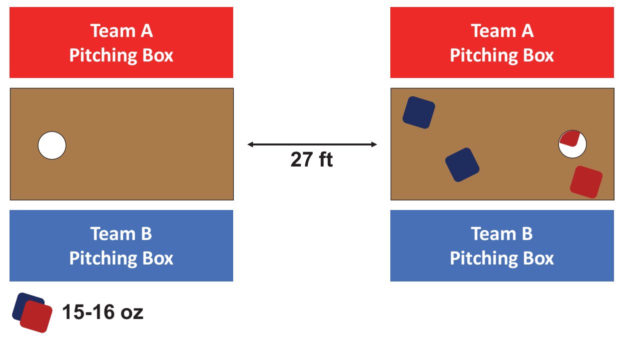


Figure 2: Court layout, viewed from above; the right board includes an example game state (foul bags not shown).

Rules of the Game

According to the American Cornhole Organization (ACO; 2016), the game can be played by 2 players (1 per team, "singles") or 4 players (2 per team, "doubles"). Once each player on a side of the court (see Figure 2) has pitched all (4) of their bags, the "frame" has ended and scoring commences: each bag that remains on the board (a "woody") gains 1 point, each bag that passes through the cornhole board hole at anytime within the frame (a "cornhole") gains 3 points, and any other bags (each a "foul") do not affect the score. It is important to note that while pitching, it is possible for one player to [un]intentionally cause a prior bag in the frame, from either team, to transition between states (e.g. an aggressive player could knock another player's bag off the board) – this is why scoring must wait till the end of a frame. Typically "cancellation" scoring is used: the points of one player cancel out the points of their opponent, and so up to one team can score per frame. For example, consider the state illustrated on the right board of Figure 2: Red earns one (1) woody and one (1) cornhole, while Blue earns two (2) woodies and zero (0) cornholes, and so 4 points - 2 points = 2 points are awarded to Red for that frame. If playing singles, the players switch court sides and repeat; in doubles, the other two players then toss from the opposite board. The game ends when a team's score reaches or exceeds 21 points.

AI Challenges for a Robotic Player

Developing a robotic cornhole player involves tackling a variety of AI problems, including ...

State Estimation What is the current state of the frame, including location of the board[2] & hole, as well as the location, color, and position of each bag[3]? This task is complicated by sensor noise, dynamic lighting conditions over the course of a game, and the frequent presence of perceptually aliased states (e.g. a bag is pitched through the hole, and is then either occluded from view or proceeds to appear as a foul behind the board).

Robotic Actuation The primary action of the robot is to reliably pitch a bag according to a desired set of parameters (e.g. velocity & angle or voltage & time delay), whereas a secondary task involves bag management (load the next bag for pitching, pick up pitched bags after a frame).

Environmental/Action Modeling Given a particular state estimate, as well as actuation parameters, what will be the [probabilistic distribution over/most likely] resulting frame state?

Decision Making Given a range of legal actuation parameters, and associated result model(s), as well as an adversarial model (e.g. minimax), what is the optimal actuation parameters in order to achieve a victory?

Adaptation There are a variety of opportunities for the robotic player to learn, both online and offline, to improve its play, including to improve state estimation (e.g. machine-learning driven vision), actuation (e.g. self-calibration), modeling (e.g. tune probabilistic estimates), and decision making (e.g. guide adversarial search).

Importantly, many of these tasks can be tackled independently, at least in part, and additionally scaled to the class focus/level and available time/resources.

Necessary Hardware and Software

The minimal equipment necessary to engage the robotics aspects of this task is a cornhole set, or at least the raw materials to construct one[4]. If purchased, sets range from desktop size (presently a 10.5" x 6" x 1.1" board costs under \$15), to "backyard" sets ($\sim$ \$40-60), to certified tournament boards (\$300). (Note that this purchase will likely have positive community-building side effects.)

As discussed in a later section, we have begun work on several of these tasks. For state estimation, we have achieved fairly high location/color accuracy with commodity webcams ($\sim$ \$70) mounted on a tripod ($\sim$ \$50). For reduced-size boards/bags, and 5–7 foot distances, we have made good progress via a student-designed, 3D-printed arm using commodity servos and controller boards ($\sim$ \$100-200; see Figure 3). However, it is likely more expensive arms or alternative approaches (e.g. bag "cannons") will be necessary to scale to regulation distances. We have yet to successfully tackle additional hardware-related complications, including mobile vision, non-vision sensors, a mobile robot to collect pitched bags, nor robotic self-managed bag inventory and pitch preparation.

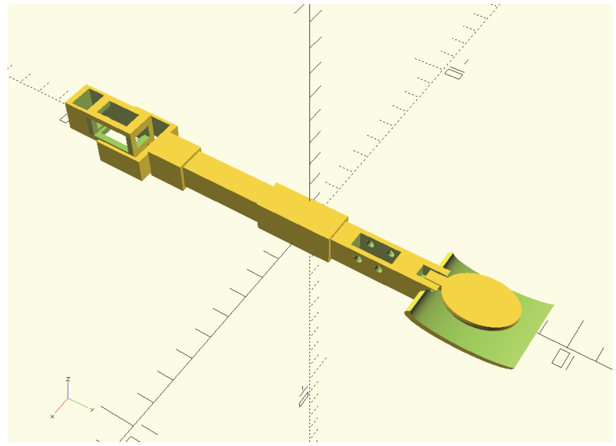


Figure 3: Student-designed, 3D-printed pitching arm.

Thus far all software has been freely available, and primarily open source, including Ubuntu Linux, Arduino IDE, OpenCV, and OpenSCAD. For vision-based projects, we have developed, and plan to freely share, a dataset of 198 annotated cornhole-frame images, as well as software to facilitate capture and annotation of new images. For fast prototyping, and algorithmic development, we foresee the desire to have a reasonably accurate 3D cornhole simulator, but have not yet developed this software.

Experience to Date

At the time of writing, we have completed three semester-long projects related to cornhole, and have two additional projects in the works. We consider ourselves in the infrastructure-building phase of this task, where students themselves iteratively build scaffolding for successive projects. The ultimate goal is a set of self-contained projects, for multiple class foci/levels, that pit students against the task of developing robotic systems that can effectively play the game of cornhole.

CHUCK

In the first project, a recently graduated electromechanical undergraduate was tasked with developing a robotic actuator for pitching in the game of cornhole. Importantly, the student was not limited in method of projection, but ended up developing CHUCK, an underhand-tossing solution, similar to human pitching (see Figure 3).

The arm is table-mounted (see Figure 4) and can toss 5-7 feet reliably. The static 3D-printed components are actuated by one Dynamixel AX-18a servo and two Savox servos:

[2] It is common for the board itself to shift position over time.

[3] Bags are typically made of a flexible material, such as canvas, twill, or synthetic suede, and filled with a particulate, such as corn, plastic pellets, or sand – estimating the three-dimensional position and fill-distribution of the bag can be crucial to making strategic pitching decisions.

[4] Primarily wood, screws, fabric, fill & saw, drill, sewing kit.

Figure 4: CHUCK, the 3D-printed arm, pitching a bag.

one "shoulder" provides directional adjustment, one "elbow" provides projection, and one "thumb" holds/releases the bag. An Arbotix-M microcontroller provides regulated power and control, and is programmed via the standard Arduino IDE. In total the arm parts can be purchased, and it can be manufactured using hobbyist 3D printers, for about $200-300, which is significantly less than many arms available for purchase.

The Arduino program running on CHUCK exposes a simple set of serial commands (i.e. an API): `graspBeanBag`, `throwBeanBag` (parameterized via voltage and release time), `rotateArm` (parameterized via angle), and `disconnect`. Given this language, it is trivial to instruct CHUCK in the language of one's choosing. As an example, we developed a Java client that not only pitches the bag, but takes as feedback a "reward" (woody=1, cornhole=3, foul=0) and, using a simple temporal-difference (TD) learning algorithm (see Equation 1; Sutton and Barto 1998), learns the correct elbow parameterization to achieve accurate throws (though currently limited to no rotation and no other bags on the board). We have released as open source[5] the CAD design for CHUCK, the Arduino source for the microcontroller, and an example Java client. Though we will certainly improve CHUCK iteratively, this progress forms a solid base for end-to-end robotic actuation and learning.

$$Q(s_{t+1}, a_{t+1}) = Q(s_t, a_t) + \alpha(r_t - Q(s_t, a_t)) \quad (1)$$

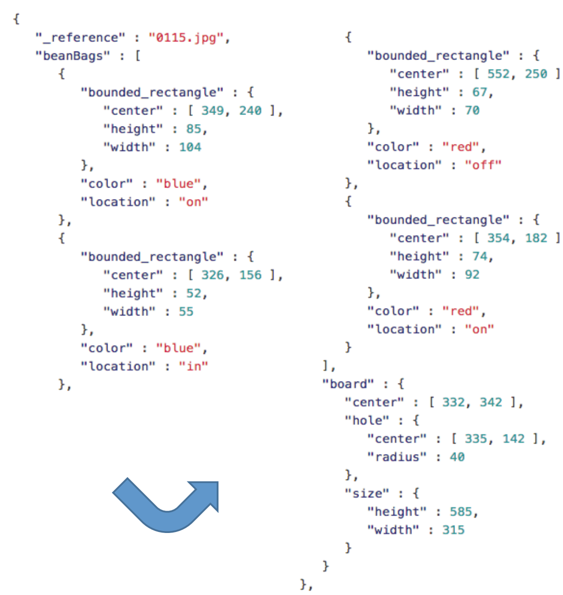
Q : Value Function ($S \times A \rightarrow \mathbb{R}$)
s : State
a : Action
t : Time
r : Reward
α : Learning Rate

Cornhole Vision Image Dataset

After working on CHUCK, the same student proceeded to develop an image dataset to facilitate future vision-based projects – the result is a set of 198 annotated images. For example, Figure 5 shows an example input image and Figure 6 shows the corresponding annotation (in JSON format). To assist in developing this dataset, the student wrote a set of software programs that visualize the annotation data, as well as help automate the process of capturing and annotating new images. In particular, the software (written in C++ with OpenCV) captures an image, performs basic object detection to "guess" at board/hole/bag locations, and provides a simple interface by which to confirm/revise these values. We have released both the dataset and the annotation/visualization software as open source[5], and invite the community to submit their own contributions, ideally including new boards/bags, lighting conditions, cameras, etc.

[5]https://github.com/cornhole

Figure 5: Image #115 from the dataset.




```
{
  "_reference" : "0115.jpg",
  "beanBags" : [
    {
      "bounded_rectangle" : {
        "center" : [ 349, 240 ],
        "height" : 85,
        "width" : 104
      },
      "color" : "blue",
      "location" : "on"
    },
    {
      "bounded_rectangle" : {
        "center" : [ 326, 156 ],
        "height" : 52,
        "width" : 55
      },
      "color" : "blue",
      "location" : "in"
    },
    {
      "bounded_rectangle" : {
        "center" : [ 552, 250 ],
        "height" : 67,
        "width" : 70
      },
      "color" : "red",
      "location" : "off"
    },
    {
      "bounded_rectangle" : {
        "center" : [ 354, 182 ],
        "height" : 74,
        "width" : 92
      },
      "color" : "red",
      "location" : "on"
    }
  ],
  "board" : {
    "center" : [ 332, 342 ],
    "hole" : {
      "center" : [ 335, 142 ],
      "radius" : 40
    },
    "size" : {
      "height" : 585,
      "width" : 315
    }
  }
},
```


Figure 6: Annotation of image #115 from the dataset.

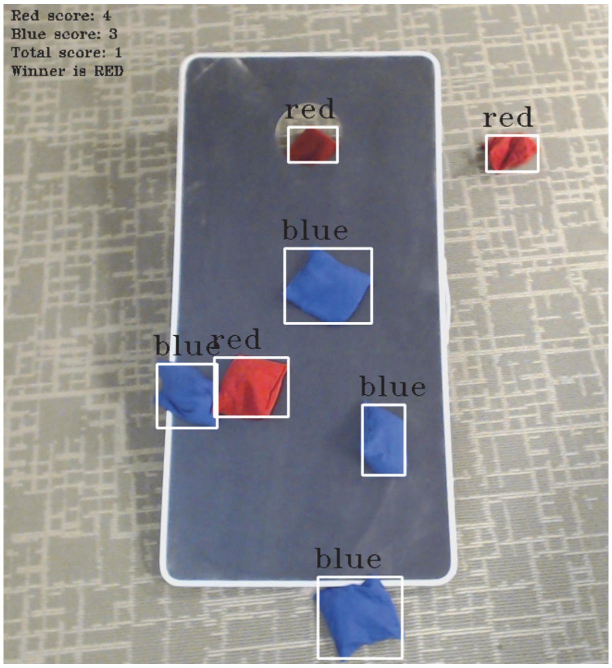


Figure 7: Example run of the Automatic Score Keeper (ASK) for cornhole.

ASK

As a first client of the vision dataset, two seniors in computer science developed the Automatic Score Keeper (ASK) for cornhole (Eshimkanov and Derbinsky 2017). Written in C++, the system used OpenCV to analyze a live video stream and, in real time, output scoring information (see Figure 7). The students found the dataset to be crucial for quickly prototyping their vision-based bag-detection algorithm – they used it to assess progress as they iterated on various heuristic [non-learning] approaches. Once the single-frame algorithm worked sufficiently (80-85% accuracy on our dataset), they enabled video and the system is reasonably accurate in empirical usage (though suffers when lighting conditions change). The students then incorporated frame-change detection to detect alternating throws (useful for estimating state), and have a functioning score keeper.

Future Plans

In this paper we presented cornhole as a compelling and adaptable AI task. Furthermore, we have contributed a number of useful pieces, including a vision dataset, which allows students to very quickly prototype and validate certain state-estimation components, as well as an inexpensive arm design for robotic actuation. However, we foresee that for cornhole to be truly accessible, there needs to be additional infrastructure-building, including . . .

Simulation For students to quickly tackle the remaining subtasks, it will be useful to have a reasonably accurate simulation system. In particular, we see this piece as necessary to engage undergraduates in earlier years, who may not have the experience to simultaneously engage the challenges of electronics/mechanics, software development, and AI. Ideally, it would be highly modular to support custom sensor and actuator noise models, as well as optionally make use of student state-estimation/decision-making code. (We look forward to a VR interface, which would not only provide a natural GUI, but add engagement value.)

Tournament Support To boost engagement, we are considering building software to support tournament play, and thereby have cornhole bots compete.

Gameplay Dataset To support decision-making components, particularly those that are learned/data-driven, we are considering building a dataset of played games. Thus, student systems could build opponent models, as well as perhaps learn useful strategies for play. This is a much larger undertaking than the vision dataset, but builds on our existing infrastructure.

We are hopeful that the community engages in this challenge and collaboratively supports the next generation of cornhole AI curricula.

Acknowledgments

We acknowledge the funding support of the Presidential EPIC Mini-Grant[6].

References

2015. American Cornhole Championships X. http://www.espn.com/espn/story/_/id/13236785/american-cornhole-championships-x. Accessed: 2016-09-10.

Eshimkanov, T., and Derbinsky, N. 2017. Automatic score keeper (ASK) for cornhole via computer vision. In *2017 AAAS Annual Meeting*.

2016. ACO Official Rules for the Sport of Cornhole. http://americancornhole.com/aco-official-rules-for-the-sport-of-cornhole/. Accessed: 2016-09-10.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

[6]https://www.wit.edu/epic