University of Bucharest
Faculty of Mathematics and Computer Science
Department of Computer Science

Iuliana Georgescu
Bogdan Alexe
MSC in Computer Science
May 2021

Computer Vision - Project 2

Video analysis in the sport of curling

Objective

The goal of this project is to develop an automatic system for video analysis of footages for the game of curling. The system should be able to detect the stones placed on the ice surface (also known as curling sheet), track the stones as they are thrown by players and infer the current score given the configurations of the stones on the ice surface.

Description of the curling game[1]

Two teams, each with four players, take turns sliding heavy, polished granite stones, also called rocks, across the ice curling sheet toward The House, a circular target area marked on the ice and segmented into four concentric circles (Figure 1). Each team has eight stones, with each player throwing two. The purpose is to accumulate the highest score for a game; points are scored for the stones resting closest to the centre of the house at the conclusion of each end, which is completed when both teams have thrown all of their stones. A game usually consists of eight or ten ends.

[1] Information taken from Wikipedia: https://en.wikipedia.org/wiki/Curling

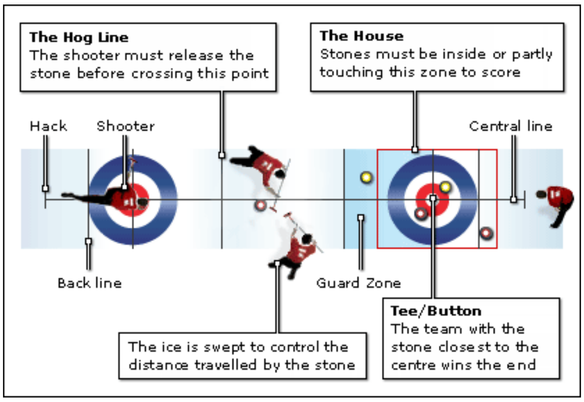

Figure 1: *Players release stones from the hog line with the goal of putting the stones in the house such that, in the end, they will win as many points as possible.*

The player can induce a curved path, described as curl, by causing the stone to slowly turn as it slides. The path of the rock may be further influenced by two sweepers with brooms or brushes, who accompany it as it slides down the sheet and sweep the ice in front of the stone. ”Sweeping a rock” decreases the friction, which makes the stone travel a straighter path (with less ”curl”) and a longer distance. A great deal of strategy and teamwork go into choosing the ideal path and placement of a stone for each situation, and the skills of the curlers determine the degree to which the stone will achieve the desired result. This gives curling its nickname of ”chess on ice”.

Data description and grading scheme

The release data directory (available at `https://tinyurl.com/CV-2021-Project2`) contains three directories: *train*, *test* and *evaluation*. The directories *train* and *test* have the same structure, although the *test* data will be made available after the deadline. The *train* directory contains data organized in four subdirectories corresponding to the four Tasks that you need to solve. The subdirectories are:

- *Task1* - this directory contains 25 training images showing the red and yellow stones on the ice surface in a constrained scenario. In this scenario, each image shows The House from above. The Task1 consists in correctly counting how many stones are on the ice surface and how many of these stones are red and yellow. The format that you need to follow is the one used in the ground-truth files (located in the subdirectory *ground-truth*) with the first line containing the total number of stones and the second and third lines containing the number of red and yellow stones.

  Grading scheme for Task1. In the test scenario we will release 25 testing images similar with the training images. By correctly solving the Task1 on all images you will get 1.5 points. You get 0.03 points/image if your algorithm counts correctly the number of stones in the image (if the first line in your output file is correct) and another 0.03 points/image if your algorithm counts correctly how many red and yellow stones are on the ice (if the second and third lines in your output file are correct).

- *Task2* - this directory contains 15 training videos in the constrained scenario showing the ice surface from above. The videos usually end when all stones have stopped from moving on the ice surface. At this point we would like to know what is the potential score based on the configuration of stones on the ice. We will follow the curling scoring rules adapted to a single video. For our purpose we will consider that only one team (Red or Yellow) can score during a video. The Team Red plays with the red stones and the Team Yellow plays with the yellow stones. The team with the most stones in The House closest to the curling bullseye (the button) is awarded points. A stone which touches the edge of The House is considered to be inside The House, so it could potentially score a point. So if, after some stones are thrown, Team Red has a stone right on the button, and Team Yellow has a stone a few feet off the button, Team Red scores a point, so the score is 1-0. If Team Red has one stone on the button and a stone a few feet off the button, while Team Yellow has a stone on the outer edge of the house, Team Red scores two points, so the score is 2-0. If Team Red has three stones outside The House and Team Yellow has two stones inside The

House, Team Yellow scores two points, so the score is 0-2. If no team has stones inside The House the score is 0-0. The format that you need to follow is the one used in the ground-truth files (located in the subdirectory *ground-truth*) with the first line containing the total number of points for Team Red and the second line containing the number of points for Team Yellow. Please pay attention to the scoring rules as you will have to output similar scores in the test phase.

Grading scheme for Task2. In the test scenario we will release 15 testing videos similar with the training videos. By correctly solving the Task2 on all images you will get 1.5 points. You get 0.1 points/video if your algorithm correctly identifies the score (if the first and second line in your output file are correct).

- *Task3* - this directory contains 15 training videos in the constrained scenario showing the ice surface from above. The task is to track the stone thrown by a player. You should track the stone from the initial frame to the final frame of the video. The stone is either red or yellow. The initial bounding box of the stone to be tracked is provided for the first frame (the annotation follows the format [xmin ymin xmax ymax], where (xmin,ymin) is the top left corner and (xmax,ymax) is the bottom right corner of the initial bounding-box). In each video we will consider that your algorithm *correctly tracks the stone* if in more (greater or equal) than 80% of the video frames your algorithm *correctly localizes the stone to be tracked*. We consider that your algorithm *correctly localizes the stone to be tracked* in a specific frame if the value of the IOU (intersection over union) beetween the window provided by your algorithm and the ground-truth window is more than 20%. The format that you need to follow is the one used in the ground-truth files (located in the subdirectory *ground-truth*) with the first line containing the number of frames N of the video, and each line having the format [frame_index xmin ymin xmax ymax]. The first frame for which we provide the bounding box initialization has frame_index 0, the last frame of a video with N frames has frame_index $N - 1$. Please note that the first line of the annotation file has the format [N -1 -1 -1 -1] as it is easy to load the entire matrix $(N + 1) \times 5$ to assess the correctness of your algorithm. Notice that for this task the stone to be tracked appears in each frame of the video.

  Grading scheme for Task3. In the test scenario we will release 15 testing videos similar with the training videos. By correctly solving the Task3 on all images you will get 1.5 points. You get 0.1 points/video if your algorithm correctly tracks the stone in a video.

- *Task4* - this directory contains 10 training videos in an unconstrained scenario. In this scenario the ice surface is seen from different viewpoints. The task is to track the stone thrown by a player. You should track the stone from the initial frame to the final frame of the video. The initial bounding box of the stone to be tracked is not provided for this task. Please notice that this task is similar to Task 3 but is harder: you are not given the initial bounding box, there are multiple viewpoints of the scene and also, it may happen, that there are several scale changes of the stone to be tracked in the entire video. The evaluation protocol is similar to the one used in Task 3. In each video we will consider that your algorithm correctly tracks the stone if in more (greater or equal) than 80% of the video frames your algorithm correctly localizes the stone to be tracked. The format that you need to follow is the one used also in Task 3 and can be observed in the ground-truth files (located in the subdirectory *ground-truth*) with the first line containing the number of frames N of the video (again in the format [N -1 -1 -1 -1]), and each line having the format [frame_index xmin ymin xmax ymax]. Notice that, different than Task 3, there could be frames in the video that do not contain the stone to be tracked. If your algorithm will output a window in these frames, this windows will be considered false positive and penalized by the evaluation protocol.

Grading scheme for Task4. In the test scenario we will release 10 testing videos similar with the training videos. By correctly solving the Task4 on all images you will get 1 point. You get 0.1 points/video if your algorithm correctly tracks the stone in a video.

- 0.5 points - ex officio. Please note that we will award the 0.5 points only to those students who submit their results in the REQUIRED format.

Evaluation

The directory *evaluation* shows how the evaluation will take place on the test data after the deadline. It contains the following subdirectories:

- *fake_test* - this directory exemplifies how the test data will be released in the *test* directory (similar with Project 1), keeping the structure of the previously described *train* directory;
- *submission_files* - this directory exemplifies the format of the results data that we expect from you to submit in the second stage. You will have to send your results in this format, uploading a zip archive of a folder similar with the one called *Alexe_Bogdan_407*. Please note that if you don't submit your results in this format you will not get the 0.5 points from ex officio;
- *code_evaluation* - this directory contains code that we will use to evaluate your results using the ground-truth data. Make sure that this code will run on your submitted files. The ground-truth data will be released after you send us your results.

Deadlines: Submit a zip archive containing ONLY your code and a pdf file describing your approach until Saturday, 26^{th} of June using the following link `https://tinyurl.com/CV-2021-PROJECT2-SUBMISSIONS`. Please do not include in your archive training images or videos. Notice that this is a hard deadline, no projects will be accepted after the deadline. Your code should include a README file (see the example in the materials for this project) containing the following information: (i) the libraries required to run the project including the full version of each library; (ii) indications of how to run the solution for each task and where to look for the output file. Students are allowed to submit solution in the format of Jupyter Notebbok files with the code being commented. This can replace the pdf file describing their approach. Students who do not describe their approach (using comments thoughout the code or a pdf file) will incur a penalty of 0.5 points.

On Sunday 27^{th} of June we will make available the test data. You will have to run your system on the test data provided by us and upload your results in the same day as a zip archive using the following link `https://tinyurl.com/CV-2021-PROJECT2-RESULTS`.