Research Article

An AI-Based Curling Game System for Winter Olympics

Xuanke Shi, Quan Wang, Chao Wang, Rui Wang, Longshu Zheng, Chen Qian, and Wei Tang 1,2

1SenseTime Research, Beijing 100080, China
2State Key Laboratory of Fluid Power and Mechatronic Systems, Zhejiang University, Hangzhou 310027, China

Correspondence should be addressed to Wei Tang; weitang@zju.edu.cn

Received 15 June 2022; Accepted 26 September 2022; Published 26 October 2022

Copyright © 2022 Xuanke Shi et al. Exclusive Licensee Science and Technology Review Publishing House. Distributed under a Creative Commons Attribution License (CC BY 4.0).

The real-time application of artificial intelligence (AI) technologies in sports is a long-standing challenge owing to large spatial sports field, complexity, and uncertainty of real-world environment, etc. Although some AI-based systems have been applied to sporting events such as tennis, basketball, and football, they are replayed after the game rather than applied in real time. Here, we present an AI-based curling game system, termed CurlingHunter, which can display actual trajectories, predicted trajectories, and house regions of curling during the games via a giant screen in curling stadiums and a live streaming media platform on the internet in real time, so as to assist the game, improve the interest of watching game, help athletes train, etc. We provide a complete description of CurlingHunter’s architecture and a thorough evaluation of its performances and demonstrate that CurlingHunter possesses remarkable real-time performance (~9.005 ms), high accuracy (30 ± 3 cm under measurement distance > 20 m), and good stability. CurlingHunter is the first, to the best of our knowledge, real-time system that can assist athletes to compete during the games in the history of sports and has been successfully applied in Winter Olympics and Winter Paralympics. Our work highlights the potential of AI-based systems for real-time applications in sports.

1. Introduction

Although AI has made a series of breakthroughs in games (such as poker [1] and Go [2]), materials science [3], chemistry [4], biology [5], mathematics [6], debate [7], and ancient text restoration [8], applying AI to sports [9–14] in real time is a challenging problem, because real-time applications in sports assist games is a domain which requires excellent real-time performance, high accuracy, and good stability, and the environment of sports is a real world that is particularly complex and has many uncertainties, which would greatly affect the performances of the AI-based systems.

The intense curling game at Winter Olympics has attracted great interest and has generated relevant researches [15–17]. As a strategic sport, curling has the reputation of "chess on ice" [18–20]. Its origin could be dated back to the 16th century in Scotland and curling has been an official sporting event of the Winter Olympics since 1998. During the sport, athletes need to pay attention to the positions of the curling stones in real time and make timely strategic adjustments based on information including stones’ actual trajectories, predicted trajectories, and house regions. At the same time, these information also significantly influence spectators’ feelings of watching games. In addition, analysis of curling stones’ motion could provide great help to athletes training and mechanical analysis of curling research. However, in the actual curling game, there is a lack of such real-time system to display these information to assist the games.

Here, we propose an AI-based curling game system, termed CurlingHunter, which can be applied in actual curling games in real time to assist athletes to compete, enhance the interest of the game, etc. Due to the regulations of curling game, no auxiliary equipment can be added to the curling stones; hence, only noncontact measurement methods such as machine vision can be used in CurlingHunter. CurlingHunter has solved these problems: (i) the problem of accurately capturing relatively small curling stones through long-sighted distance (>20 m) in the superlarge space environment with many occlusions; (ii) the problem of lens distortion correction in large scenes without interfering with the ice tracks; (iii) the problem of accuracy of curling stone’s visual positioning on the ice sheet; (iv) the problem of
occlusions which would interfere with tracking and accuracy, while curling stone is easily blocked by athletes wiping ice, other peoples, or objects during games; (v) the problem of tracking and reidentifying multiple curling stones due to that all curling stones have identical appearance features; and (vi) the problem of runtime in sing-camera system and multicamera system. As the first system to be applied to a curling game, CurlingHunter demonstrated excellent performances in 2022 Beijing Winter Olympic Games of curling and 2022 Beijing Winter Paralympic Games of curling. Although we focus on curling, our system is readily transferable to other sports.

2. Results

2.1. System Architecture. The curling game of the 2022 Beijing Winter Olympics was held in Beijing “Ice Cube” (Figure 1(a)), which was the largest curling stadium in the history of the Olympics. There were four ice tracks about 46-meter length and 5-meter width in the middle of “Ice Cube” (figure S1). Our CurlingHunter consisted of forty-two cameras arranged in “Ice Cube” (Figure 1(a), figure S2, and Materials and Methods) with overlapping field of views (Figure 1(b), figures S3–S6, and Materials and Methods) to ensure that every part of ice tracks was captured by at least three cameras from different angles so as to solve the problem of occlusions including people, truss, and camera. The cameras were arranged on three heights, i.e., 2nd floor of grandstand (F2), Cat walk (CW), and Truss, which were distributed around the ice tracks, including two types with different functions, i.e., speed dome camera and box camera (Figure 1(c)). A large screen with 170 square meters was placed in east side of the stadium to display the house regions and curling stones’ actual trajectories and predicted trajectories of the four ice tracks in real time (Figure 1(a)), which could help athletes to make a preliminary judgment and develop a strategy during the game and make watching games more interesting. Two types of curling stones, red and yellow (figure S7), were used in curling game, each with identical appearance.

Due to many uncertainties in practical curling game, it is a huge and complex task to reconstruct curling stones’ actual trajectories in real time, predict its future trajectories, and analyze its motion parameters. These processes involve utilizing single-camera tracking, multicamera fusion, lens distortion correction, deep learning, etc. Considering the variety of tasks required to tackle these problems, it seems infeasible to envisage a monolith solution in the form of an end-to-end system, such as a single deep learning network. Instead, CurlingHunter was designed to break these problems into modular tangible tasks. It is interesting that some of these tasks are proposed in this practical curling game, such as optimal tracking strategy for small targets in large environment, visual positioning of targets on the ice, image distortion correction in large scene, and real-time multicamera data association, promoting the proposal and application of some new methods in AI field. In the following, we succinctly introduce all main modules of CurlingHunter and how they handle the information from the previous module. More details of these parts are described in the Materials and Methods.

CurlingHunter consists of three main modules (Figure 1(d)): single-camera tracking and visual positioning, multicamera data association and trajectory generation, and motion analysis and trajectory prediction. The first module is based on single-camera processing, which is multi-threaded and executed simultaneously. Through the first module, the data association of multi-frame information of forty-two cameras can be obtained, but the tracklets generated by a single view is easily affected by short-term or long-term occlusions. In the second module, we design a time synchronization to synchronize the tracklets of each camera generated in the first module at the same time, then propose a long short-term matching mechanism (LSTMM) by curling stones’ locations and history trajectory information to assign the global curling ID in spatial dimension and match long short-term historical information in time dimension, and lastly, utilize multicamera fusion to reconstruct curling stones’ actual trajectories in real time. In the third module, we take advantage of the asymmetric weighted least-square (AWLS) to calculate the velocity, acceleration, and angular velocity of curling stone in real time and propose a model based on the long short-term memory (LSTM) to predict the curling stones’ future trajectories in real time.

2.1.1. Single-Camera Tracking and Visual Positioning. Single-camera tracking and visual positioning is achieved through three stages (Figure 2(a)). The first stage is single-camera detection and tracking. Unlike the general tracking-by-detection paradigm, our method mainly focuses on the local target template patch (LTPP), a target-centered image patch which is defined as the smallest carrier of target information that reflects the target appearance features and location information, which can reduce the overhead of a lot of computing resources brought by frame-by-frame detection because a large amount of background information in the image is meaningless for tracking. RetinaNet [21] with multiscale feature pyramid network (FPN) [22] is used to detect curling stones and find their bounding boxes (BBox), so as to obtain the original LTPP. A refine module (details are described in the Materials and Methods) is proposed to update and optimize the original LTPP so as to obtain refined LTPP with high quality. To track the curling stones, we take inspiration from Siamese-RPN [23, 24] and design a lightweight Deep Siamese Tracker (details are described in the Materials and Methods). However, tracking by historical templates cannot meet the need to track new appearing objects in the scene and would be affected by long-term occlusion. We can get new appearing LTPP by the detection module which is called at a certain frequency. The less the detection module is called, the faster the system will be. In order to maximize the performance of the system, our detection module is performed on the whole image every 10 frames, which can achieve good tracking performance and low time overhead. To avoid the exchange of curling ID which has already been tracked, a greedy matching strategy is used to match the currently detected refined LTPP.
with the historically tracked refined LTTP, and we only use ClIoU [25, 26] of the BBoxes of curling in the refined LTTPs to estimate the similarity of curling stones across frames. By combining detection and tracking with refine module, we achieve a robust and efficient tracking process handling complex and changeable real environments, which is suitable for solving the problem of accurately capturing relatively small curling stones through long-sighted distance (>20 m) in the superlarge space environment.

In the second stage, we design a landmark detection network (details are described in the Materials and Methods) to obtain the landmark coordinates of curling stones, so as to get the real position of curling stones in LTTP. Usually, the center of the detection frame is used as the position center of the target, but this method is not suitable for curling stones. For cameras with different viewing angles and different pixel positions of the same camera in a large scene, the perspectives of the curling stone vary greatly, resulting in a completely different positions between the center of the detection frame and the corresponding position of the curling stone. Therefore, if the center of detection frame is used as the position of curling stone, it would cause large systematic errors in global positioning of curling stone. To overcome this problem, we define a curling landmarks consisting of handle head of stone, handle tail of stone, and bottom center of stone, denoted as \( h = (h_a, h_b, h_c) \).
Figure 2: Overview of each module. (a) Overview of single-camera tracking and visual positioning. The module consists of three stages, i.e., sing-camera detection and tracking, landmark detection, and lens distortion correction. (b) Overview of multicamera data association and trajectory generation. The module consists of three components, i.e., homography projection, time synchronization, LSTMM-based region growing, and multicamera fusion module. (c) Overview of trajectory prediction. The framework consists of three components, i.e., encoder, rotation fusion module, and decoder.
(Figure 2(a)). This method determines position of curling stone as curling landmarks instead of the detection frame center that is time variable, which ensures that curling stone is in the same position from different camera angles, avoiding positioning errors caused by fuzzy definition. We use deep learning to get coordinates of curling landmarks and corresponding accuracy scores.

However, the coordinates of landmarks are affected by image distortion. In the field of machine vision, the distorted image can be rectified by camera calibration [27] using a checkerboard to calibrate the camera, but camera calibration in large scenes is very difficult, and the size and quantity of the checkerboard cannot meet requirements of calibration. Especially in the curling game, the checkerboard cannot be arranged in ice tracks due to that the ice tracks would be interfered by checkerboard. In this stage, to solve the problem of checkerboard method failure, we take examples from algebraic methods [28, 29] and propose a fully automatic lens distortion correction method (figure S8 and Materials and Methods), which is based on the structured straight-line elements in images. The method constructs an appropriate energy function for the linear elements in the image and utilizes nonlinear optimization to iteratively correct the distorted linear elements, so as to complete lens distortion correction. By the method, we do not need to calibrate intrinsic parameters of cameras and can easily get the distortion correction model through only a single image in a large scene. Through the distortion correction model, we can easily obtain accurate coordinates of curling stones.

2.1.2. Multicamera Data Association and Trajectory Generation. All curling information tracked by the single-camera system is fed into the multicamera system, as shown in Figure 2(b), and the position of curling stone in the world coordinate system is a bridge for different cameras. To take advantage of the complementary gains of each camera, we use the world coordinates of curling stones to integrate the information of each camera. The coordinates of curling stones in the image of each camera can be projected on the ice track using proposed homography projection (details are described in the Materials and Methods), so as to complete the transformation from the image coordinate system to the world coordinate system. The world coordinates of curling stones are stored in the queue of each camera. In order to avoid the time asynchronous interference and simultaneously obtain the curling position information of each camera, we adopt a time synchronization algorithm (details are described in the Materials and Methods) to complete the timestamp alignment of each camera. And then the time-aligned curling stones’ information of each camera at the reference timestamp is obtained, which is the base to associate the tracklet information from different cameras.

There are two challenges in multicamera tracking: one is that each camera only covers a part of the curling stones, resulting that the total number of curling stones is difficult to determine and the total number that can be seen from multiple cameras may vary in the time dimension; the other is that it is hard to associate the information across different cameras due to that all curling stones have identical appearance features. In addition, it is a well-known NP-hard problem to tackle the data association problem across cameras, and the runtime of the algorithm grows exponentially as the number of cameras grows; existing methods are mostly offline and cannot meet the requirements of real-time data processing. A long short-term matching mechanism (LSTMM) (table S1 and Materials and Methods) based on regional growing algorithm is proposed to assign the global curling ID in spatial dimension and match long-short term historical information in time dimension to overcome these problems. With the help of accurate spatial temporal curling positioning and the local appearance information from tracklets, the problem of frequent ID switch in single-camera tracking caused by short-term occlusion can be solved.

In some cases, the curling stone cannot be captured and tracked by any camera due to occlusion or other reasons. However, the reidentification technique cannot be used since it would mislead the matching process due to that all curling stones have identical appearance features. To enhance performance of multicamera tracking in long-term occlusion, a global spatial temporal matching mechanism (details are described in the Materials and Methods) by bigraph matching is proposed. With the long-term matching mechanism, the region growing algorithm can be executed along the time dimension and curling stones which have been occluded for a long time can find the corresponding ID. Curling stones are merged from different cameras which have a same global curling ID. To improve the accuracy of multicamera sensor fusion, we use the accuracy confidence scores as a guide for weighted fusion of curling coordinates. Finally, global trajectories of curling stones can be generated in real time by multicamera sensor fusion.

2.1.3. Motion Analysis and Trajectory Prediction. The kinematic and mechanical analysis of curling [15, 30] is difficult and complex. The characteristics of the ice surface would be affected by a series of factors, such as temperature, humidity, and athletes rubbing the ice, resulting in some local or overall small changes in the ice surface, which would even affect the final results of curling games. High-quality curling motion data is of great significance for the mechanical analysis of curling motion and the study of ice surface characteristics. The asymmetric weighted least square method (figure S9 and Materials and Methods) combined with high-frequency motion capture based on forty-two cameras is used to calculate the velocity, acceleration, and angular velocity of curling stone in real time, which can reflect the quality of the ice surface and help athlete train.

Uncertainty in the overall and local behavior of the ice sheet have brought great challenges to modeling curling motion, and the frictional force of the ice changes essentially with each throw. In addition to the uncertainty of the ice surface, the ice rubbing by athletes and multicamera measurement errors are also difficult to model by physical models. All of these factors make the precise future state of curling intrinsically unpredictable. Although the motion modeling of curling contains many uncertainties, the motion behaviors of curling in the future can be approximated. The motion of curling approximately satisfies the Markov
assumption [31], so we can adopt sequence model to predict the curling stone’s future trajectory. We introduce an encoder-decoder framework (details are described in the Materials and Methods) based on long short-term memory (LSTM) network [32] which predicts the future trajectory based on curling’s partial observation in a throw. The framework of our trajectory prediction model is shown in Figure 2(c), which consists of three key components: encoder, rotation fusion module, and decoder. Through the model, CurlingHunter can obtain curling stone’s predicted trajectory in real time, which can help athletes judge and enhance the interest of the game.

2.2. Evaluation and Applications. To comprehensively evaluate the performances of CurlingHunter, we conducted detailed experiments and applied it in actual curling games. First of all, we tested the effectiveness and positioning accuracy of each module, evaluated the overall real-time performance, and compared with the existing AI systems used in sport games to verify that only CurlingHunter could be applied in real time during the games and be broadcast live, while other existing AI systems could not achieve. Finally, we presented the applications of CurlingHunter in actual curling games, including 2021 Wheelchair Curling World Championships, 2022 Beijing Winter Olympics, and 2022 Beijing Winter Paralympics.

2.2.1. System Evaluation. To verify the effectiveness of single-camera tracking and minimize the time and resource overhead of single-camera tracking, we conducted relevant verification based on the actual situation. For quantitative evaluation, we adopted IDF1, MOTA, and MOTP for tracking performance which was widely accepted by CLEAR MOT metrics [33]. At the same time, we used FPS (frame per second) to measure the time overhead of the program. As shown in Figure 3(a), the runtime of our method increases gradually with the number of targets. Nonetheless, our method always outperforms frame-by-frame detection schemes (such as SORT), since the image contains a large number of invalid regions by using frame-by-frame detection schemes. In addition to the time advantage, our method can bring a smaller resource overhead, since using one graphics card per video is a luxury in practical applications. As shown in Figure 3(b), the frame-by-frame detection scheme (such as SORT) produces little changes with the increase of the number of objects in the scene, because the detection is carried out on the whole image. As the number of processed videos increases, it is difficult to guarantee real-time performance with limited resources when using the full-image tracking scheme. However, our method only using the search region near the target template for tracking task is better than frame-by-frame detection, which is instructive for the deployment of tracking algorithms with limited resources. As shown in Figure 3(c), as the detection interval increases, the IDF1 of our method becomes better, MOTA becomes worse, and MOTP becomes better. To strike a balance between performance and time, we set the detection interval to ten, in which case our method performs better than SORT algorithm that does not use the appearance feature of target. To verify the validity of our refine module, we designed an ablation experiment. As shown in table S2, the high-quality target template refined by refine module help us to get better performance than original SiamsRPN++. To verify the accuracy of our visual positioning, we evaluated the measurement accuracy of five points at base camp with known coordinates for each ice track. As shown in Figure 3(d), the proposed landmark detection has a large accuracy improvement over the center coordinate of BBox, and the error is further reduced by lens distortion correction, thereby further improving the accuracy and achieving 30 ± 3 cm under measurement distance > 20 m. Movie S1 restores the actual trajectory to the ice track so as to visually verify the accuracy of CurlingHunter. High accuracy guarantees that CurlingHunter can be used in actual curling games.

We used IDF1 and ID-Switch to quantitatively evaluate the performance of multicamera tracking to reconstruct trajectories and analyzed the effects of long-term matching mechanism, short-term matching mechanism, and single-camera tracking performance of each camera on multicamera tracking and reconstruction of trajectories, respectively. The ablation experiments were conducted on the multiview videos of a curling game, where 12 videos covered a complete track and lasted about 15 minutes. The ID switch rate could reflect bad case in tracking caused by occlusion or other reasons. As shown in Figure 3(e), long short-term matching mechanism is the best in both metrics. As shown in Figure 3(f), we actively introduce per-camera ID switch probabilities ranging from 0% to 35%, although the probability of per-camera local ID switch is large enough, we find that the global ID switch is still relatively small. Our method is robust in complex and varied real-world environments due to that we combine single-camera tracking information and curling motion information for long-term short-term matching rather than relying only on a single submodule.

We evaluated the results of velocity and angle measurements in a wheelchair curling game where no athletes rubbed the ice. As shown in figure S10, the noise of velocity and angle calculation due to measurement error can be eliminated in real time, so we can obtain the monitoring of the motion information of curling during the curling movement without attaching any additional equipment. As shown in Figure 3(g), our method predicts future trajectories better than those estimated by Kalman filtering.

To verify the real-time performance of CurlingHunter, we conducted an overall evaluation of the runtime of each module (tables S3–S4, and Materials and Methods). Unlike the usual researchers discussing a specific method, we focus on how to comprehensively utilize each module to realize that the whole performance better than parts. We tested the overall runtime of CurlingHunter in a large number of actual curling games, and the average runtime is ~9.005 ms, the time lag of which human eyes cannot distinguish, demonstrating that CurlingHunter can be applied for actual curling games in real time.

Existing AI systems are used in tennis [9], basketball [10], and football [11], which are mainly used for postgame analysis to help athletes train or assist the referee in judging the games, and cannot be applied in real time to assist
games. The Hawk-Eye System used in tennis is the most mature and advanced technologies applied in sports, but its runtime is \(~\sim 10\) s, which is above 1,000 times slower than ours (our CurlingHunter only takes \(~\sim 9.005\) ms). Table S5 compares CurlingHunter with existing AI systems in detail, demonstrating that CurlingHunter is the first AI sports system in history that can be applied in real time to assist the game and improve the interest of watching game, etc.

### 2.2.2. Winter Olympics

We conducted many practical tests on CurlingHunter, where movie S2 shows one of the tests, and invited relevant professionals to evaluate the results. The results show that CurlingHunter possesses remarkable real-time performance (\(~\sim 9.005\) ms), high accuracy (30 \pm 3 cm under measurement distance > 20 m), and good stability, proving that CurlingHunter can be used in actual curling games. We applied CurlingHunter to the 2021 Wheelchair Curling World Championships from October 23rd, 2021 to October 30th, 2021, 2022 Beijing Winter Olympic Games of curling from February 10th, 2022 to February 17th, 2022, and 2022 Beijing Winter Paralympic Games of curling from March 5th, 2022 to March 15th, 2022, and achieved excellent results, which obtained highly commended from the World Curling Federation (WCF) President Kate Caithness, athletes, coaches, referees, spectators, commentators, media, etc.

The giant screen with an area of 170 square meters, as shown in Figure 4(a), displays the four house regions and the curling stones’ trajectories in real time at a 1 : 1 ratio. Figure 4(b) shows the actual applications of CurlingHunter
in Winter Olympics and Winter Paralympics, where athletes “watching the giant screen” during the games has become the norm. The curling ice tracks are very long, resulting that athletes are dozens of meters away from the house regions, so it is difficult to know where the target is located dozens of meters away and how close to the center of the house region in the past. In addition, in the past, athletes could only rely on memory for the trajectory of each throw, and how to correct the next throw could also only depend on memory. CurlingHunter solves these problems technically. By watching the giant screen, athletes could clearly know the actual positioning of curling stones, the actual trajectory, the predicted trajectory, the specifics of the current throw, and how to correct the next throw, which greatly liberate the memory of athletes, so as to better assist athletes in curling games.

Through the live video streaming of CurlingHunter (Figure 4(c), movie S3, and movie S4), the spectators can clearly see the trajectory of each throw by the athletes. In the past, the spectators could only watch a partial perspective of the live broadcast and did not know the curling in other three ice tracks or the overall situation of the curling game, but CurlingHunter presents to spectators the most intuitive and comprehensive display, significantly enhancing the experience of watching the games. Figure S11 and movie S5 show curling stone’s velocity, acceleration and rotation angle in real time during the games. In addition, we developed a management system (Figures 4(d)–4(f) and figure S12) for CurlingHunter to record and manage the trajectory, motion analysis, and ice surface path for each curling game. The management system can save all the game matches and their related information, including the team of the game, the person who threw the curling stone, the direction of the game, time, temperature, and humidity. The trajectory management system (Figure 4(d)) can dynamically display the trajectory of the curling stone; the motion management system (Figure 4(e)) can dynamically display the velocity, acceleration, and rotation angle of the curling stone; and the ice surface path management system (Figure 4(f)) can dynamically display the friction degree of the ice surface during the curling movement. The management system can be used for game

![Figure 4: 2022 Beijing Winter Olympics Games and 2022 Beijing Winter Paralympics. (a) The giant screen with an area of 170 square meters. (b) The actual applications of CurlingHunter in Winter Olympics and Winter Paralympics. (c) The live video streaming of CurlingHunter in Winter Olympics and Winter Paralympics. (d) The trajectory management system. (e) The motion analysis management system. (f) The ice surface path management system.](image-url)
management, postmatch analysis, and postmatch training for athletes, etc. Interestingly, the actual motion data of the curling games can be used for mechanical analysis of curling research.

3. Discussion

In this work we developed CurlingHunter, a curling game system based on a series of AI technologies, with remarkable real-time performance (~9.005 ms), high accuracy (30 ± 3 cm under measurement distance > 20 m), and good stability. CurlingHunter has been successfully applied to actual curling game, filling in the gaps of the systems which are utilized to assist curling game in real time. CurlingHunter is the first, to the best of our knowledge, real-time system that assist athletes to compete during the games in the history of sports and successfully applied in Winter Olympics and Winter Paralympics. The achievements described in this work represent a major milestone in the development of AI technologies applied in the real world and promote the development of curling games. In addition, CurlingHunter offers a new platform for further extending to other sports and using to academic research of multi-target multi-camera tracking.

4. Materials and Methods

4.1. Curling Game. Curling, as a combination of bowling and chess [31], is a turn-based game in which two teams play alternately on the ice tracks. There are four ice tracks in curling game, where each ice track consists of side line, house region, hack, tee line, and hog line, as shown in figure S1. There are eight athletes in the two teams, and there are usually ten round games. A curling game requires two sets of curling stones, where each set consists of eight curling stones. Different from the ice surface of figure skating or short track speed skating, the ice track surface of curling game is not completely flat, whose top layer is covered with specially made tiny particles; hence, athletes need to sweep the ice surface to change the friction between the curling stone and the ice surface so as to adjust the direction. As shown in figure S7, the diameter, height, and weight of curling stone are 30 cm, 11.43 cm, and 19.96 kg, respectively.

4.2. Positions and Layouts of Cameras. The forty-two cameras are divided into three heights, i.e., 2nd level. The forty-two respectively. The forty-two cameras are divided into three heights, i.e., 2nd level. The forty-two respectively. The forty-two cameras are divided into three heights, i.e., 2nd level. The forty-two respectively.

4.3. Coverage Areas of Cameras. The forty-two cameras overlap and cover all areas of the ice tracks, ensuring that every part of the ice tracks are covered by at least three cameras with different viewing angles to solve the problems of occlusion. Each camera has a jurisdictional area, and the actual monitor area of each camera is larger than its jurisdictional area. Cameras 1-4 and 9-12 in F2 overlap each other and cover two tracks in the north and south, respectively, ensuring that every part of the ice tracks is covered, where their jurisdictional areas are shown in figure S3A and actual monitor areas are shown in figure S4. Cameras 5-8 in F2 overlap each other and cover far and near ends of two tracks in the north and south, respectively, ensuring that every part of the ice tracks are covered, where their jurisdictional areas are shown in figure S3B and actual monitor areas are shown in figure S4. Cameras 16, 17, 20-22 and 26, 27, 30-32 in CW overlap each other and cover the first track in the north and south, respectively, ensuring that every part of the ice tracks A and D is covered, where their jurisdictional areas are shown in figure S3C and actual monitor areas are shown in figure S5. Cameras 15, 18, 19, 23, 24 and 25, 28, 29, 33, 34 in CW overlap each other and cover the second track in the north and south, respectively, ensuring that every part of the ice tracks B and C is covered, where their jurisdictional areas are shown in figure S3D and actual monitor areas are shown in figure S5. Cameras 13, 14 in CW cover near ends of two tracks in the north and south, respectively, where their jurisdictional areas are shown in figure S3E and actual monitor areas are shown in figure S5. Cameras 35-42 cover 8 house region, respectively, where their jurisdictional areas are shown in figure S3F and actual monitor areas are shown in figure S6.

4.4. Single-Camera Tracking and Visual Positioning

4.4.1. Detection. For generation of the initial LTTP, RetinaNet [21] with multiscale feature pyramid network (FPN) [22] is used to detect curling stones and find their bounding boxes (BBox). FPN adopts top-down architecture with skip connections, which can produce a single high-level feature map with fine resolution. We detect the curling stones on the finest layer combining the high-level and low-level semantics, which is useful for accurate localization of small curling stones as it has less secondary sampling of the original image. However, there are a lot of problems in the original LTTP generated by detection, such as exiting false positives, jitter of the curling stone' bounding boxes, and poor quality of the local target template patch. To solve these problems, a refine module is proposed to update and optimize LTTP.
4.4.2. Refine Module. The refine module consists of two branches, regression branch and confidence branch, which are used to optimize the coordinate of the original LTTP in the image and estimate the quality of final LTTP. For original LTTP, if LTTP slightly deviate from the ground truth, the LTTP coordinate in the image could be adjusted by regression branch; if LTTP greatly deviate from the curling or does not contain any curling stone by confidence branch, the LTTP could be discarded.

The training of the refine module consists of two stages. Firstly, we train the regression branch end to end using stochastic gradient descent (SGD) with momentum. During the training, we augment the data by applying random occlude, rotate, horizontal flip, and so on, which can significantly enhance the generalization and robustness of the neural network for complex scenes. To make the regression easier to converge, we make the proportion of curled fixed as $\alpha = 1.2$ and normalize LTTP as same size. The loss for regression is smooth $L_1$ loss. The center coordinate of LTTP is $(x_c, y_c)$, and the refine bounding box in LTTP is $(w_c, h_c)$. We can directly regress the values of $(\delta_x, \delta_y, \delta_w, \delta_h)$ to rectify the coordinate in the original patch $(x_c, y_c, w_c, h_c)$ due to that the patch is normalized. Ground truth in LTTP can be denoted as $(g_1, g_2, g_3, g_4)$. The smooth $L_1$ loss is

$$\text{smooth}_{L_1}(x, \sigma) = \begin{cases} \frac{1}{2} \sigma^2 x^2, & |x| \leq \frac{1}{\sigma^2}, \\ |x| - \frac{1}{2\sigma^2}, & |x| > \frac{1}{\sigma^2}. \end{cases}$$

The regression loss is

$$L_{\text{reg}} = \sum_{i=0}^{3} \text{smooth}_{L_1}(\delta_i - g_i, \sigma).$$

Secondly, the confidence branch shares the same feature extraction subnetwork with regression branch. We freeze the weighted value of the regression branch when it converges. To quantify the quality of LTTP more accurately and remove the false positive detection, we directly regress the confidence score of LTTP. The quality score of LTTP can be approximated truncated CIoU [25, 26] between bounding boxes from detection result $D$ and result $\hat{D}$ from the regression branch. To balance the positive and negative samples, we collected a large amount of low quality LTTP and set its score to zero. The truncated CIoU is

$$\text{confidence}(D) = \begin{cases} \text{CIoU}(D, \hat{D}), & \text{CIoU}(D, \hat{D}) > 0, \\ 0, & \text{CIoU}(D, \hat{D}) \leq 0, \end{cases}$$

where confidence($D$) ranges continuously from 0 to 1.

During the inference phase, the two branches simultaneously output the refined bounding boxes and the quality score of LTTP. The low-quality LTTP or false detection would be removed if the confidence score is less than $\text{thr}_{\text{low}}$. Finally, the quality of LTTP is improved by the refine module.

4.4.3. Tracking. The proposed lightweight Deep Siamese Tracker consists of a Siamese subnetwork for feature extraction and a region proposal subnetwork for proposal generation. For feature extraction, the template branch encodes the historical target patch improved by proposed the refine module, the detect branch encodes the search patch which contains the region in current frame where the target patch in previous frame was located. For region proposal, the template feature and the search patch feature can be associated by a correlation operation. Then, the region proposal network [34] is adopted to regress the coordinate of the target proposal and finish the foreground-background classification. We get new LTTP in next frame after tracking, to improve the quality of LTTP, the refine module is used to improve the LTTP and estimate its confidence score which can remove the illegal LTTP. However, the number of objects in the scene changes dynamically, and new target may appear in the field of view at any time; we can get new appearing LTTP by the detection module which is called at a certain frequency as a supplement. To avoid the exchange of curling ID which has already been tracked, a greedy matching strategy is used to match the currently detected refined LTTP with the historically tracked refined LTTP. We only use CIoU [25, 26] of the BBoxes of curling in the refined LTTPs to estimate the similarity of curling stones across frames because all the curling stones with same color have an identical appearance feature. When the similarity is greater than 0.5, the ID of the curling keep the same as the historical tracking information; otherwise, we assign the tracking ID to the newly appearing curling. By the skillful combination of detection and tracking with the refine module, we achieve a robust and efficient tracking process handling complex and changeable real environments which is suitable for tracking multiple small targets in large scenes.

4.4.4. Visual Positioning. The landmark detection is processed on the refined LTTP. As shown in Figure 2(a), we define curling landmarks which consist of landmarks of stone handle head, stone handle tail, and stone bottom center, denoted as $h = (h_a, h_b, h_c)$. When the rectified curling proposal is given, we detect the landmarks of curling to accurately improve accuracy of visual measurement. The network architecture of curling landmark detection is similar to the refine module, so we directly regress the coordinate of curling landmark in the normalized LTTP, and the landmark accuracy score is also given by the confidence branch. For each landmark $\tilde{h}_i \in h$, the landmark predict accuracy score is defined as

$$\text{confidence}(\tilde{h}_i) = \begin{cases} 1 - \frac{\sigma(\tilde{h}_i, \tilde{h}_i)}{\tilde{h}_a - \tilde{h}_b}, & \frac{\sigma(\tilde{h}_i, \tilde{h}_i)}{\tilde{h}_a - \tilde{h}_b} \leq 1, \\ 0, & \frac{\sigma(\tilde{h}_i, \tilde{h}_i)}{\tilde{h}_a - \tilde{h}_b} > 1, \end{cases}$$

where $\sigma$ is the smooth CIoU [25, 26].
which is given by $\| \tilde{h}_t - h_t \|$, which distributes between 0 and 1. The more accurate the landmark regression accuracy is, the closer the confidence score is to 1; otherwise, it is closer to 0. When the confidence score is greater than 0.75, the system is optimum. Usually, we find the position landmark is more robust than handle landmark due to that the former is the overall feature of curling stone while the latter is the local feature of curling which is easily affected by occlusion. With the accuracy score for each curling in each camera, the problem of landmarks’ inaccurate estimation could be improved due to occlusion or other reasons.

4.4.5. Lens Distortion Correction. The distortion model can be defined as

$$
(x - x_c) = L(r) \left( \begin{array}{c} x_d - x_c \\ y_d - y_c \end{array} \right),
$$

where $(x, y)$ is the coordinate in the corrected image, $(x_d, y_d)$ is the coordinate in the original image, and $(x_c, y_c)$ is the center of the lens distortion model. We define $r$ as the distance from image point to the center of the lens distortion model. It can be calculated as

$$
r = \sqrt{(x_d - x_c)^2 + (y_d - y_c)^2}.
$$

$L(r)$ determines the distribution of the image distortion which is given by

$$
L(r) = k_0 + k_1 r^2 + k_2 r^4.
$$

A fully automatic lens distortion correction method is proposed. Firstly, we adopt the improved Hough Line Transform [28] to detect the distorted lines in the image. We denote $k_c$ by the tuple $(k_0, k_1, k_2, x_c, y_c)$ which defines the distortion model. The undistorted image point $x_{ij}$ is $j$th point of line $i$ in the undistorted image by using equation (5) to rectify the original image point. By distortion model $k_c$, the line $j$ in the undistorted image is formulated as

$$
\cos (a_j^k x) + \sin (a_j^k y) + d_j^k = 0.
$$

Then, an iterative nonlinear optimization is performed by minimizing the average of the square distance from the corrected image points to the corresponding line. The energy function is given by

$$
E(k_c) = \frac{\sum_{i=1}^{N_c} \sum_{j=1}^{N_j} \cos (a_j^k x_{ij}) + \sin (a_j^k y_{ij}) + d_j^k}{\sum_{i=1}^{N_c} N_{ij}}.
$$

We fix the center point of lens distortion $(x_c, y_c)$ and adopt Levenberg-Marquardt algorithm to minimize the energy function as below:

$$
k_c^{n+1} = k_c^n - (\nabla^2 E(k_c^n) + \lambda I)^{-1} \nabla E(k_c^n).
$$

By the automatic lens distortion correction method, we do not need to calibrate intrinsic parameters of cameras and can easily get the distortion correction model through only single image in a large scene. And the curling landmarks could be corrected by using equation (5). By this method, accurate visual positioning in large scenes becomes feasible, which is important for cross-camera association.

4.5. Multicamera Data Association and Trajectory Generation

4.5.1. Homography Projection. For $N$ cameras with different views, we use $L_t$ to denote the tracking and positioning results in the same batch of video frames received at time $t$:

$$
L_t = \{ (L_{1t}, T_{1t}), (L_{2t}, T_{2t}), \cdots, (L_{Nt}, T_{Nt}) \},
$$

where $(L_i^t, T_i^t)$ represents the information of $i$th camera at time $T_i^t$. For each tracklet in $L_i^t \in L_t$, $i = 1, \cdots, N$. We use homography matrix $H_i$ to transform the coordinate from image plane of $N$th camera to the world plane. As shown in Figure 2(b), the curling trajectories of each camera are generated synchronously by homography projection. We use $G_t$ to denote the same batch trajectories of all the cameras:

$$
G_t = \{ (G_1^t, T_1^t), (G_2^t, T_2^t), \cdots, (G_N^t, T_N^t) \},
$$

where $G_i^t$ contains all the curling trajectories of $i$th camera and $T_i^t$ represents the corresponding timestamp.

4.5.2. Time Synchronization. The timestamps from different cameras are not strictly aligned. For the timestamps $T_i^t$ and $T_j^t$, $i, j = 1, \cdots, N$, $i \neq j$, they are usually not equal numerically. In some extreme cases, they could also be significantly out of sync. Therefore, we store the trajectory data of different cameras in their respective queue data structures, and use the queue head data of different cameras as the same batch $G_t$. When the biggest timestamp interval is larger than the $T_{thr}$, we dequeue the trajectory data with the smallest timestamp and enqueue the data of the next frame until the same batch of data satisfies that the timestamp interval is less than the $T_{thr}$, where $T_{thr} = 33$ ms.

To avoid the time asynchronous interference of each camera, we linearly interpolate the coordinates of the curling along the time dimension. As shown in Figure 2(b), $T_{thr}$ is the earliest timestamp in the same batch. We use linear interpolation for trajectories of other cameras to get the trajectory information at time $T_{thr}$.

$$
G_i^t = G_i^0 \times \frac{T_{thr} - t}{t_1 - t_0} + G_i^1 \times \frac{t - t_0}{t_1 - t_0},
$$

where $t_1$ and $t_0$ are the timestamps of the earliest and the latest frame in the batch respectively.
where $t$ is the timestamp to be interpolated. $\widetilde{G}_t$ is used to denote the aligned local trajectories:

$$
\widetilde{G}_t = \left\{ \left(G_t^1, G_t^2, \ldots, G_t^N, \tilde{T}_t \right) \right\},
$$

(14)

where $G_t^i$ is the curling trajectories of $i$-th camera at aligned timestamp $\tilde{T}_t$.

With the time synchronization algorithm, the time-aligned curling stones’ information of each camera at the reference timestamp is obtained, which is the base to associate the tracklet information from different camera.

4.5.3. Multicamera Tracking. The proposed LSTMM is outlined in Table S1. For a single view, the tracklet can pass the GCID information to the next frame. We denote the initial curling seeds by set $S_t$ which collects the local spatial temporal association from all cameras at $T_t$ timestamp. We select the initial seeds which have been assigned GCID by heuristic information of each camera tracking result. For each curling $c_t^j \in \widetilde{G}_t$, we compute the pairwise euclidean distance between the curling $c_t^j$ without GCID and the seed curling $c_t^i$ belongs to initial seeds set $S_t$. The growth criteria is as follows:

$$
\left\| c_t^j - c_t^i \right\|_2 < \text{dist}_{\text{thr}}, j = 1, \ldots, M_t, k = 1, \ldots, K,
$$

(15)

where $M_t$ is the total number of curling stones in the aligned trajectory $\widetilde{G}_t$, $K$ is the total number of initial seeds in set $S_t$, and dist$_{\text{thr}} = 30$ cm which is equal to the diameter of a curling stone. As shown in Figure 2(b), when the condition is satisfied, the same batch curling stones from different views should be gradually clustered to be a same GCID identity in the world plane, until the same batch of curling stone from different views is traversed. With the help of accurate spatial temporal curling positioning and the local appearance information from tracklet, the problem of frequent ID switch in single-camera tracking caused by short-term occlusion can be solved.

Most of the curling stones in the cameras have been assigned a GCID through the single-camera tracking and the short-term matching mechanism. However, owing to the complexity of the real-world environment, the curling identities are often occluded by the objects, e.g., athletes, trusses in the air. We design a spatial temporal long-term matching mechanism to compensate for the limitations of short-term matching. We construct a bipartite graph between the historical trajectories and the curling stones without GCID for each view. To reduce the solution space of this problem, we design an elimination criteria to exclude solutions which are impossible. Firstly, for each camera, we remove the trajectories where GCID have been assigned in this view. Secondly, we remove the impossible matching from current to history trajectory. To improve the reuse rate of the latest state of the trajectory, we estimate the curling stones’ motion state and trajectory tangent equation of each trajectory at the latest moment. For a certain curling stones’ trajectory at $T_t$ timestamp, we choose the $Q$ stones’ coordinate in the trajectory which are closest to time $T_t$, where the stones’ coordinate at time $T_t$ is $c_t = (x_t, y_t)$. The trajectory can be represented by a quadratic equation as follows:

$$
x = a_1 x^2 + a_2 y + a_3,
$$

(16)

where $a_1, a_2$, and $a_3$ are the coefficients of the equation. For this least squares problem $X = AY$. We use normal equation to get the optimal solution.

$$
A = (Y^T Y)^{-1} Y^T X.
$$

(17)

By the trajectory equation, the trajectory tangent equation at $(x_t, y_t)$ is

$$
k_1 x + k_2 y + k_3 = 0,
$$

(18)

where $k_1 = 1, k_2 = -2a_1 y_t + a_2$, and $k_3 = k_2 y_t - x_t$. In order to determine whether the curling stone is in a stationary state. We construct a curling stone’s motion status matrix $S$ from the latest $Q$ zero-centered curling stone’s coordinates. By singular value decomposition (SVD), we have $S = U \Sigma V^T$, where the singular value are $\sigma_1$ and $\sigma_2$ which can reflect the variance of the curling stone’s coordinate in the direction of columns vector of $V^T$. When the singular value $\sigma_1$ and $\sigma_2$ are smaller than $\sigma_{\text{thr}}$, we set $a_1 = a_2 = a_3 = 0$ to represent the curling stone is stationary in the past. Based on the above formula, the second elimination criteria is as follows:

$$
\left\{ \begin{array}{l}
\left\| c_t^{j+1} - c_t^i \right\|_2 < \text{dist}_{\text{thr}}, \\
\left| k_1 x_t^{j+1} + k_2 y_t^{j+1} + k_3 \right| < \text{dist}_{\text{thr}}, \quad \text{if } a_1 = a_2 = a_3 = 0,
\end{array} \right.
$$

(19)

For each curling stone $c_t^{j+1} \in C_t^{j+1}$, if the $j$th curling stone in the $i$th camera has not been assigned a GCID and meet the matching elimination criteria, we remove the matching relationship in bipartite graph.

By the elimination criteria, the solution space of the bipartite graph matching is greatly reduced. Then, we find the optimal assignment solution by Hungarian algorithm. With the long-term matching mechanism, the region growing algorithm can be executed along the time dimension and curling stones which have been occluded for a long time can find the corresponding ID.

For these curling stones which have not been assigned the GCID by LSTMM-based region growing, we assign new GCIDs to newly appearing curling stones. Because of the incremental ID allocation strategy, we can detect the legality of appearing curling stone which can help us largely eliminate false detections. For each camera, the new curling stones’ GCID information, which are potential curling seeds for regional growing algorithm, can also be passed across frames.
Finally, curling stones are merged from different cameras which have a same global curling ID. To improve the accuracy of multicamera sensor fusion, we use the accuracy confidence scores as a guide for weighted fusion of curling coordinates. The accuracy confidence score of each curling stone in different cameras can be learned by convolution neural network. It can be calculated by

$$c_i^m = \frac{\sum_{i=1}^{N} w_{ij}q^j}{\sum_{i=1}^{N} w_{ij}},$$  \(20\)

where all the merging curling stones $c_i^j$ have the same GID across cameras and $N$ is the number of cameras that the curling can be observed simultaneously at the moment. The accuracy confidence scores, denoted as $w_{ij}$, can be calculated by equation [4]. Through multicamera sensor fusion, we get the real-time global trajectories of curling stones.

4.6. Motion Analysis and Trajectory Prediction

4.6.1. Motion Analysis. High-frequency motion capture can help us to analyze the motion of curling stone all the time, but measurement noise has a big impact on this problem, especially in multicamera sensor systems. We store the time, but measurement noise has a big impact on this problem. We can get the velocity and acceleration by the angle between the curling handle and the $x$ axis in the world coordinate system. Meanwhile, to reduce the influence of accumulated measurement errors of $L_t$, which is perpendicular to the trajectory direction, we project the current curling position coordinates to the tangent direction of the curling trajectory. To approximate the motion of curling at time $T$, we use asymmetric weighted least-square (AWLS) method along time domain to locally solve this problem. The sampling points can be approximated by

$$E_{\Omega T}(t) = \sum_{i=1}^{\mu} p_i(t)q_i(t) = g^T(p(t), \forall t \in \Omega_T),$$  \(21\)

where $g^T(t)$ is a $\mu$-dimensional basis function vector, $p(t)$ is a coefficient vector which needed to be estimated, $\Omega_T$, is an asymmetric local neighborhood $[T_{t-1}, T_{t+1}]$ around the moment $T$. To estimate the velocity and rotation of curling stone, we set $t_1 = 50$ and $t_2 = 10$ which can real-time output of results delayed by 10 frames. We set $q(t) = [1, t, t^2]^T$ with its order number $\mu = 3$. The coefficient vector $p(t)$ can be solved by minimize the following weighted least square errors:

$$E_{\Omega T} = \sum_{T_{t-1}}^{T_{t+1}} w_{T_t} \left[ F^T(t) - g^T \right],$$  \(22\)

where $g^T$ is the actual observed value. It can be formulated as a vector matrix formulation:

$$E_{\Omega T} = (Qp - g)^T W (Qp - g),$$  \(23\)

where

$$Q = \begin{bmatrix} 1 & T_{t-1} & T_{t-1}^2 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & T_{t+2} & T_{t+2}^2 \end{bmatrix},$$

$$g = \begin{bmatrix} g_{t-1} \\ \vdots \\ g_{t+2} \end{bmatrix},$$

$$W = \begin{bmatrix} w_{T_{t-1}} & 0 & \cdots & 0 \\ 0 & w_{T_{t-1}+2} & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & w_{T_{t+2}} \end{bmatrix}.$$  \(26\)

By minimizing the error function, we can obtain the coefficient vector $p = (p_1, p_2, p_3)$ by

$$p = (Q^T W Q)^{-1} Q^T W g.$$  \(27\)

We can also get the velocity and acceleration by the first derivative and the second of $F^T(t)$ with respect to $t$.

The optimizations process for curling stone’s angle $\theta$, and distances $L_t$ are slightly different in terms of weights. For the fitting problem of $L^2(t)$, the weight at $T_t$ is calculated by

$$w_{T_t \mid L} = \frac{1}{\sqrt{2\pi\sigma_t}} e^{(-T_{t-1}^2-\sigma_t^2)/2\sigma_t^2} \frac{1}{\sqrt{2\pi\sigma_v}} e^{(-T_{t-1}-\sigma_v^2)/2\sigma_v^2},$$  \(28\)

where $T_{t-1} \in \Omega_T$, $\sigma_t$ is the time distance variance, and $\sigma_v$ is the velocity distance variance. The temporary speeds $V_t$ is given by

$$V_t = \frac{L_{t-1}}{T_{t-1}-T_{t-2}}.$$  \(29\)

For $\theta^2(t)$ function estimation, the weight at $T_t$ is calculated by

$$w_{T_t \mid \theta} = \frac{1}{\sqrt{2\pi\sigma_t}} e^{(-T_{t-1}^2-\sigma_t^2)/2\sigma_t^2} \frac{1}{\sqrt{2\pi\sigma_v}} e^{(-W_{t-1}-\sigma_v^2)/2\sigma_v^2},$$  \(30\)

where $\sigma_v$ denotes the angular velocity variance. The temporary speeds $W_t$ is given by

$$W_t = \frac{\theta_t - \theta_{t-1}}{T_{t-1}}.$$  \(31\)

By using the asymmetric weighted least-square (AWLS), we can robust estimate the velocity, acceleration, and angular velocity of curling stone in real time, where these data can reflect the quality of the ice surface and help athlete train.
4.6.2. Trajectory Prediction. The properties of the ice surface change with temperature, humidity, etc. To simply this problem, we assume that the overall performance of the ice remains constant during each throw; therefore, the motion of curling can approximately satisfy the Markov assumption which we can adopt sequence model to predict the curling stone’s trajectory.

The long short-term memory (LSTM) \([32]\) network which is a variant of recurrent neural networks has been proven to be very successful for sequence prediction task \([35–37]\) such as speech recognition, machine translation, and human trajectory prediction. We introduce an encoder-decoder framework based on LSTM which predicts the future curling stone’s trajectory \(Y = \{Y_0, Y_1, \cdots, Y_{\text{pred}}\}\) based on curling stone’s spatial observation \(X = \{X_0, X_1, \cdots, X_{\text{obs}}\}\) in a throw. The framework of our trajectory prediction model is shown in Figure 2(c). Our model consists of three key components: encoder, rotation fusion module, and decoder. The encoder learns the physical properties of the ice surface and the motion pattern of curling stones from partial observations \(X = \{\Delta X_0, \Delta X_1, \cdots, \Delta X_{\text{obs}}\}\). Firstly, we use multilayer perceptron (MLP) to get the fixed length spatial embedding \(e_i\) of relative motion pattern \(\Delta X_i = X_i - X_{i-1}\). Then, spatial embedding can be used as input by a LSTM cell of the encoder. We define the \(i^{\text{th}}\) observation as \(X_i = (x_i, y_i)\), the \(j^{\text{th}}\) prediction as \(Y_j = (x_j, y_j)\). The encoder at \(i^{\text{th}}\) observation can be defined as follows:

\[
e_j = \psi(\Delta x_j, \Delta y_j, W_{ee}),
\]

\[
hd_i = \text{LSTM}(hd_{i-1}, e_j, W_{encoder}),
\]

where \(\psi(\cdot)\) is the spatial embedding function, \(W_{ee}\) is the embedding weight, and \(W_{encoder}\) is the weight of LSTM cell.

The trajectory of curling stone is close to a straight line when it is just thrown. The lateral movement of the curling stone caused by the rotation is even smaller than the measurement error of the system. The rotation angle is extracted from the handle landmarks of the curling stone, which is a local feature of the curling stone and is easily affected by the occlusion of the curling stone. Therefore, we use the rotation direction instead of the rotation angle to obtain a more robust evaluation. Therefore, we design a rotation fusion module to merge the rotation direction information into observation hidden state to get the hidden state \(hd_{\text{obs}}\).

To keep trajectory prediction consistent with past trajectory observation during a throw, we initialize the state of decoder by \(hd_{\text{obs}}\), where the hidden state \(hd_{\text{obs}}\) contains the assessment of the ice surface at this casting. The decoder at \(j^{\text{th}}\) observation can be defined as follows:

\[
e_j = \phi(\Delta x_{j-1}, \Delta y_{j-1}, W_{ed}),
\]

\[
h_j^d = \text{LSTM}(h_j^d, e_j, W_{decoder}),
\]

\[
\Delta X_j = \varphi(h_j^d),
\]

where \(\phi(\cdot)\) is the spatial embedding function in decoder, \(W_{ed}\) is the embedding weight, \(W_{decoder}\) is the weight of LSTM cell in decoder, and \(\varphi(\cdot)\) is the MLP function.

We embed the coordinate as a 16-dimensional vector. The dimensions of the hidden state for encoder and decoder are 32. We train the model by minimizing \(L_1\) loss by an Adam optimizer, which can minimize the deviation of the predicted trajectory from the actual ground truth. In the inference stage, we can predict the trajectory by partial observation of a whole trajectory, the prediction result can maintain the same motion patterns as observation in the same ice surface conditions.

4.7. Performance Tests. All the tests were conducted on the Intel(R) Xeon(R) Gold 6226R CPU @ 2.90 GHz and Tesla T4 GPU.

4.7.1. Runtime Tests. A video with ten targets was used for testing performance of the single-camera tracking module. RetinaNet [21] with multiscale feature pyramids network (FPN) [22] was used to generate initial LTTP firstly, and then, the refine module was used to improve the quality of the LTTP. The input image was resized by our system from 2160 × 3840 to 864 × 1536, where the size of target was 32 × 32, because it was time consuming if the full image was detected while the targets in the image were small. To speed up the runtime of the refine module and curling landmark detection on refined LTTP, we took multiple targets patch from single view as a same batch input and feed into the network. In the refine module, the target patch was cropped from the original image and the ratio of the target in the patch was fixed as 1.2. The initial LTTP was resized to 112 × 112, so as to obtain the refined LTTP. In the landmark detection, the target was cropped from the refined LTTP where the proportion of curling stone was 1.1. Similarly, the LTTP was resized to 112 × 112 and then fed into the landmark detection network. As shown in table S2, the process took about 7.598 ms, where detection took 6 ms, the refine module took 0.839 ms, and the landmark detection took 0.759 ms. In the tracking stage, the refined LTTP associated the spatial temporal information across frames, where the proportion of the target in template patch and search patch was about 1/2 and 1/3, respectively. The template branch encoded the historical refined LTTP which was resized to 63 × 63, and the detect branch encoded the search patch which was resized to 95 × 95 in current frame. Then, the refine module was used to improve quality of the LTTP and avoid false detections. After that, we cropped the target from the refined LTTP and fed the local target patch into the landmark detection network. As shown in table S3, the whole process took 5.169 ms, including 3.571 ms to update the LTTP, 0.839 ms to improve quality of the LTTP, and 0.759 ms to detect landmarks of curling. Our distortion model was a model from the distortion image to the rectified image, resulting that the image point could be corrected quickly in 0.001 ms without iterative distortion correction.

For multicamera tracking module, we used all the cameras of a single track for offline testing, which consisted of 12 cameras. We set \(\text{dist}_{\text{thr}} = 300\) mm, \(v_{\text{thr}} = 3\) m/s, while limiting the region allocated by GCID where the newly
appearing curling could only appear in the starting area of the curling throw. The cross-camera data association for 12 cameras took 0.1711 ms.

To test the overall performance of CurlingHunter in real world, we used multithreading to test, where each video was processed in parallel using one thread and the cross-camera tracking used a single thread. In order to enhance the utilization of the graphics card, we used Tesla T4 GPU to process the data of three videos at the same time. We combined detection, tracking, and the refine module where the detection interval was 10 frames. Although the number of curling stones that appeared at different times and different cameras was different, the overall time overhead of CurlingHunter maintained at only ~9.005 ms per batch of video frames. In a word, the millisecond-level processing speed paves the way for real-time applications of CurlingHunter in curling games.

4.7.2. Motion Tests. We evaluated the results of speed and angle measurements in a wheelchair race where no athletes rubbed the ice. To reduce the influence of accumulated measurement errors which was perpendicular to the trajectory direction, we projected the current curling stone’s position coordinates to the tangent direction of the curling stone’s trajectory. At the same time, we used asymmetric weighted least-square (AWLS) method along time domain to locally solve this problem where we used the next 10 frames motion information and historical 50 frames motion information to smooth the velocity of the current frame. We set \( \sigma_\nu = 1.2 \text{s} \) and \( \sigma_w = 0.5 \text{m/s} \), as shown in Figure 3(a), the noise in the velocity calculation due to measurement errors was be eliminated in real time. As shown in Figure 3(b), angular smoothing was similar to speed smoothing, where we set \( \sigma_\psi = 1.2 \text{s} \) and \( \sigma_\omega = 0.3 \text{rad/s} \).

4.7.3. Trajectory Prediction Tests. We collected a portion of the curling motion data measured by the multicamera system and randomly divided the training set and the validation set according to a certain proportion. To verify the validity of our modeling of curling stone’s trajectory prediction, we used a 3-second motion pattern of curling stone to predict the trajectory of curling stone in the next 9 seconds, where the overall state of the ice surface and the motion pattern of curling stone could be estimated approximately from 3-second observational trajectory. The hidden state encoded by LSTM could encode the motion pattern of the curling stone to help us predict curling stone’s trajectory in the future. Meanwhile, we designed a rotation fusion module to enhance the effect of rotation on future trajectory prediction. We calculated the cumulative distance error of the curling trajectory. As shown in Figure 3(g), our method predicts future trajectories better than those estimated by Kalman filtering. Among them, the error of curling stone’s trajectory prediction mainly comes from the measurement error of the observation trajectory and the uneven distribution state of the ice surface.

Data Availability

All data are available in the main text or the supplementary materials.

Disclosure

All authors are applying for patents related to the described work. The data and video in the work are authorized or publicly available.

Conflicts of Interest

The all authors declare that they have no competing financial interests.

Authors’ Contributions

X. Shi, Q. Wang, C. Qian, and W. Tang proposed and supervised the project. X. Shi, C. Wang, R. Wang, and L. Zheng designed and evaluated CurlingHunter. X. Shi, W. Tang, and C. Wang analyzed the data. W. Tang and X. Shi wrote the manuscript. All authors participated in discussions of the research and revisions of the manuscript.

Acknowledgments

We thank China National Aquatics Center for providing site and installing equipment. We thank the International Olympic Committee (IOC) and the World Curling Federation (WCF) for authorizing the project. The National Key Research and Development Program of China grant 2020YFF0304300 supported this study.

Supplementary Materials

Figure S1: the size of the curling ice tracks. Figure S2: positions and layouts of forty-two cameras. Figure S3: jurisdictional areas of forty-two cameras. Figure S4: actual monitor areas of 12 cameras in F2. Figure S5: actual monitor areas of 22 cameras in CW. Figure S6: actual monitor areas of 8 cameras in Truss. Figure S7: red curling stone and yellow curling stone in ice track. Figure S8: the examples of lens distortion correction. Figure S9: multicamera velocity analysis. Figure S10: motion analysis. Figure S11: motion analysis of curling games in real time. Figure S12: the management system. Table S1: LSTM. Table S2: ablation experiments of the refine module. Table S3: runtime tests of single-camera detection process. Table S4: runtime tests of single-camera tracking process. Table S5: comparison of existing AI systems for sports and CurlingHunter. Movie S1: restoring the actual trajectory to the track so as to verify the accuracy of CurlingHunter. Movie S2: performance test of CurlingHunter. Movie S3: the broadcast live of CurlingHunter in 2022 Beijing Winter Olympics. Movie S4: the broadcast live of CurlingHunter in 2022 Beijing Winter Paralympics. Movie S5: motion analysis. (Supplementary Materials)

References

[1] M. Moravčík, M. Schmid, N. Burch et al., "DeepStack: expert level artificial intelligence in heads-up no-limit poker," Science, vol. 356, no. 6337, pp. 508–513, 2017.
[2] D. Silver, A. Huang, C. Maddison et al., "Mastering the game of Go with deep neural networks and tree search," Nature, vol. 529, no. 7587, pp. 484–489, 2016.
[3] V. Tshitoyan, J. Dagdelen, L. Weston et al., “Unsupervised word embeddings capture latent knowledge from materials science literature,” Nature, vol. 571, no. 7763, pp. 95–98, 2019.

[4] S. H. M. Mehr, M. Craven, A. I. Leonov, G. Keenan, and L. Cronin, “A universal system for digitization and automatic execution of the chemical synthesis literature,” Science, vol. 370, no. 6512, pp. 101–108, 2020.

[5] M. Baek, F. DiMaio, I. Anishchenko et al., “Accurate prediction of protein structure and interactions using a three-track neural network,” Science, vol. 373, no. 6557, pp. 871–876, 2021.

[6] A. Davies, P. Velickovic, L. Buesing et al., “Advancing mathematics by guiding human intuition with AI,” Nature, vol. 600, no. 7887, pp. 70–74, 2021.

[7] N. Slonim, Y. Bilu, C. Alzate et al., “An autonomous debating system,” Nature, vol. 591, no. 7850, pp. 379–384, 2021.

[8] Y. Assael, T. Sommerschield, B. Shillingford et al., “Restoring and attributing ancient texts using deep neural networks,” Nature, vol. 603, no. 7900, pp. 280–283, 2022.

[9] N. E. Owens, C. Harris, and C. Stennett, “Hawk-eye tennis system,” in 2003 international conference on visual information engineering VIE 2003, pp. 182–185, Guildford, UK, 2003.

[10] K.-C. Wang and R. Zemel, “Classifying NBA offensive plays using neural networks,” in MIT Sloan Sports Analytics Conference, Boston, MA, USA, 2016.

[11] J. Fernandez and L. Bornn, “Wide open spaces: a statistical technique for measuring space creation in professional soccer,” in MIT Sloan Sports Analytics Conference, Boston, MA, USA, 2018.

[12] A. J. Moshayedi, A. S. Roy, A. Kolahdooz, and Y. Shuxin, “Deep learning application pros and cons over algorithm,” EAI Endorsed Transactions on AI and Robotics, vol. 1, pp. 1–13, 2022.

[13] A. J. Moshayedi, S. K. Sambo, and A. Kolahdooz, “Design and development of cost-effective exergames for activity incrementation,” in 2022 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE) 2022, pp. 133–137, Guangzhou, China, 2022.

[14] A. J. Moshayedi, Z. Chen, L. Liao, and S. Li, “Kinect based virtual referee for table tennis game: TTV (Table Tennis Var Sys- tem),” in 2019 6th International Conference on Information Science and Control Engineering (ICISCE) 2019, pp. 354–359, Shanghai, China, 2019.

[15] V. Honkanen, M. Ovaska, M. J. Alava, L. Laurson, and A. J. Tuononen, “A surface topography analysis of the curling stone curl mechanism,” Scientific Reports, vol. 8, no. 1, p. 8123, 2018.

[16] T. Kameda, D. Shikano, Y. Harada, S. Yanagi, and K. Sado, “The importance of the surface roughness and running band area on the bottom of a stone for the curling phenomenon,” Scientific Reports, vol. 10, no. 1, p. 20637, 2020.

[17] T. Herzog, J. Swanenburg, M. Hupp, and A.-G. M. Hager, “Effect of indoor wheelchair curling training on trunk control of person with chronic spinal cord injury: a randomised controlled trial,” Spinal Cord Series And Cases, vol. 4, p. 26, 2018.

[18] T. Ito and Y. Kitasei, “Proposal and implementation of “digital curling”,” in 2015 IEEE Conference on Computational Intelligence and Games (CIG), pp. 469–473, Tainan, Taiwan, 2015.

[19] N. W. Stewart and C. Hall, “The effects of cognitive general imagery use on decision accuracy and speed in curling,” Sport Psychologist, vol. 30, no. 4, pp. 305–313, 2016.

[20] K. Lee, S.-A. Kim, J. Choi, and S.-W. Lee, “Deep reinforcement learning in continuous action spaces: a case study in the game of simulated curling,” in Proceedings of the Thirty-Fifth International Conference on Machine Learning (ICML), pp. 2943–2952, Stockholm, Sweden, 2018.

[21] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 2, pp. 318–327, 2020.

[22] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2117–2125, Honolulu, Hawaii, USA, 2017.

[23] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, “Siamrpn ++: Evolution of siamese visual tracking with very deep networks,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4282–4291, Long Beach, California, USA, 2019.

[24] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, “High performance visual tracking with siamese region proposal network,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 8971–8980, Salt Lake, Utah, USA, 2018.

[25] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, “Generalized intersection over union: A metric and a loss for bounding box regression,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 658–666, Long Beach, California, USA, 2019.

[26] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, “DistanceIoU loss: faster and better learning for bounding box regression,” 2019, https://arxiv.org/abs/1911.08287.

[27] Z. Zhang, “A flexible new technique for camera calibration,” IEEE Transactions on pattern analysis and machine intelligence, vol. 22, no. 11, pp. 1330–1334, 2000.

[28] D. Santana-Cedrés, L. Gómez, M. Alemán-Flores, A. Salgado, and L. Alvarez, “An iterative optimization algorithm for lens distortion correction using two-parameter models,” Image Processing On Line, vol. 6, pp. 326–364, 2016.

[29] D. Santana-Cedrés, L. Gómez, M. Alemán-Flores, A. Salgado, and L. Alvarez, “Invertibility and estimation of two-parameter polynomial and division lens distortion models,” SIAM Journal on Imaging Sciences, vol. 8, no. 3, pp. 1574–1606, 2015.

[30] M. Shegelski, R. Niebergall, and M. A. Walton, “The motion of a curling rock,” Canadian Journal of Physics, vol. 74, no. 9-10, pp. 663–670, 1996.

[31] K. J. Kostuk, K. A. Willoughby, and A. P. Saedt, “Modelling curling as a Markov process,” European Journal of Operational Research, vol. 133, no. 3, pp. 557–565, 2001.

[32] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[33] K. Bernardin and R. Stiefelhagen, “Evaluating multiple object tracking performance: the clear mot metrics,” EURASIP Journal on Image and Video Processing, vol. 2008, Article ID 246309, 10 pages, 2008.

[34] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: towards real-time object detection with region proposal networks,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137–1149, 2017.

[35] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, “Social lstm: Human trajectory prediction in...
crowded spaces,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 961–971, Las Vegas, Nevada, USA, 2016.

[36] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, “Social gan: Socially acceptable trajectories with generative adversarial networks,” in Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2255–2264, Salt Lake, Utah, USA, 2018.

[37] A. Graves and N. Jaitly, “Towards end-to-end speech recognition with recurrent neural networks,” in International conference on machine learning, pp. 1764–1772, Beijing, China, 2014 Jun 18.