REAL-TIME TRACKING OF HOCKEY PLAYERS WITH AMATEUR VIDEO TRACKING

Dillon Clapp, Christopher Murphy, Adam Romlein, and Erin Zanni

Clarkson University

ABSTRACT

Current work using optical flow for sports analysis is centered around applied machine learning for player tracking or post-game analysis on pre-recorded video. This research presents a real-time application intended to be both lightweight for deployment to mobile devices and responsive to poor-quality video from a smartphone. This application targets fans and coaches looking to extract real-time velocities of hockey players, without being limited by casual camera work or low-resolution feeds. The algorithm employs the Lucas-Kanade pyramidal implementation of optical flow to identify features, and clusters the features using the mean-shift algorithm. A velocity vector for each cluster is overlaid onto the video in real-time. This research provides a strong baseline for the expansion of lightweight video processing, with room for future work to employ more complex analysis via edge computing systems.

***Index Terms*—** optical flow, Lucas-Kanade, hockey, real-time, smartphone

1. INTRODUCTION

Clarkson University is a small private research university located in rural northern New York that offers undergraduate, master, and doctoral degrees. The NCAA considers Clarkson a division III school, however, both men's and women's hockey teams are division I. Both hockey teams compete in the Eastern College Athletic Conference, a consortium of over 300 colleges in the eastern part of the United States. The women's team was established in 2003 and in that short time has won three NCAA titles; 2014, 2017, and 2018. They have won all three NCAA titles the school holds and are the first team in St. Lawrence County to win an NCAA ice hockey title. Hockey is the cornerstone of the community at Clarkson and in the surrounding area. The men's games are highly attended and on any given weekend the stands are full. However, the women's game attendance is sparse despite their recent titles. The women's games are less attended because they do not have the premium time slot like the men's team. The difference in rules and style of play: they play with more skill and are less physical. In the past, attendance has been incentivized by awarding pizza to the residence halls with the best attendance to women's games. Most recently, students can attend the women's game and keep their seats for men's game. To add to the game's excitement and enhance the fan experience, a program was created that would later be transitioned to a mobile platform.

During the game a smartphone camera can be used to find the velocity of the players. The application is easy to use and processes quickly. The user can see velocity displayed as arrows with magnitudes in relation to speed and, when paused, can see numerical data on speed. The ability to have live data on players allows fans to be more engaged and may add to the action of the women's games, thereby growing the fan base.

2. OPTICAL FLOW

The basis of this program for real-time velocity estimation of hockey players is optical flow; it determines relative motion between objects and the viewer. More specifically, optical flow finds the apparent velocity of brightness patterns between two images [1]. It can give information about the spatial arrangement of objects and the rate of change for the arrangement [1]. An important aspect of optical flow is that the presence of discontinuities can help improve the accuracy. For example, if a person walks past a white background while the camera is also panning, the perceived speed of the person is distorted. The person will either be perceived as moving faster when the camera is panning opposite of their motion, or slower when the camera is panning in the apparent direction of their motion. Background discontinuities provide additional information on camera motion which makes the person's speed more accurate.

There are many ways to calculate optical flow. In this program, Lucas-Kanade algorithm was used. Originally proposed in 1981 by Lucas and Kanade, it is considered the original image alignment algorithm [2]. The algorithm aligns one frame to the previous frame and outputs a vector containing pixel coordinates. The goal of the Lucas-Kanade algorithm is to minimize the sum of squared error between the two frames [2]. Figure 1 summarizes the Lucas-Kanade algorithm process.

The second frame is warped using an estimation of the warp in step 1 and in step 2 which is then subtracted from the previous frame resulting in the error image. In step 3, the frame gradient is warped, and in step 4, the Jacobian is computed. The warped frame gradient and Jacobian are combined to produce the steepest descent images. The steepest descent is an extension of Laplace's method from approximating an integral. In step 6, the Hessian matrix is found from the steepest descent image.

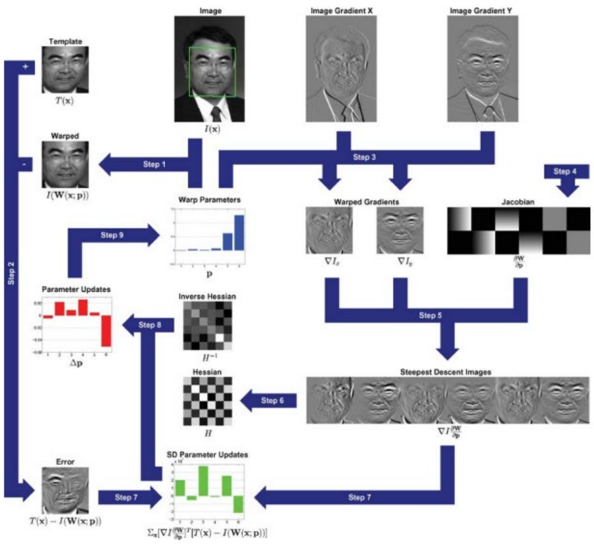

Fig. 1. A schematic overview of the Lucas-Kanade algorithm [2].

The Hessian matrix describes the curvature of a function with many variables. In step 7, the dot product of the error image from step 2 and the steepest descent images are calculated to find the updated steepest descent parameters. In step 8, the Hessian matrix is inverted and multiplied by the updated steepest descent parameters to produce the final parameter updates which are added to the parameters in step 9 [2].

Besides optical flow, the Lucas-Kanade algorithm can be used for image tracking, parametric and layered motion estimation, mosaic construction, medical image registration, and face coding [2].

3. PROGRAM DEPENDENCIES, LIBRARIES, AND LANGUAGES

The program is written in Python 3 and takes advantages of multiple third-party libraries that allow the system to function. OpenCV [3], scikit-learn (sklearn) [4], and NumPy [5] were used in the creation of the program. OpenCV handles the computer vision aspect of the program. A Python 3 implementation of the Lucas-Kanade optical flow algorithm is provided. Homography, used to determine speed, and feature identification are two other major features that the OpenCV library provides. The program makes use of the mean-shift clustering algorithm provided by scikit-learn to cluster data points. The clustering is used to group hockey players as they move across the ice. The NumPy library is useful for data and data structure manipulation. Given the vast amount of data that will need to be processed, this library provides the tools to easily manage the data. The use of these libraries ensures that the implementations of complex algorithms are correct and are bug free.

4. PROGRAM STRUCTURE

For testing purposes, the program operates on a video file that is specified at runtime via command line arguments. The video file is then used to retrieve frame data. In general, the algorithm performs optical flow between a sliding reference frame and the current frame. A grayscale filter is applied to each frame to reduce the dimensionality of the image from three to one per pixel.

The goal of the algorithm is to annotate a video feed with velocity vectors pertaining to the movement of hockey players on the ice. All operations for a given frame use the current frame and the immediately preceding frame, called the reference frame. To begin, the OpenCV library function `goodFeaturesToTrack` is applied to the reference frame to identify strong corners in the image [6]. Optical flow, provided by OpenCV, is performed on a reference frame and the current frame to predict the displacement vectors of the corners. Then, optical flow is performed in reverse, starting with the terminal points of the displacement vectors in the second frame and working backwards toward the reference frame. The result is a prediction of where the original corners should be in the reference frame. Since the corners of the reference frame is known, outliers are discarded between the known reference corners and the predicted reference corners. The presence of outliers may be attributed to noise in the image, resulting in ghost corners and poor corner flow detection.

The next step uses the results of optical flow to identify flow outliers: points with dissimilar movement are marked by an OpenCV implementation of homography employing RANSAC, or Random Sample Consensus [7]. Figure 2 is an example output from this intermediate processing step, where the red points are the outliers discovered by homography using RANSAC, while the displacement vectors and warped image overlay are a result of the dual-stage optical flow algorithm.

Green points exhibit similar movement between frames and therefore represent the background. Red points are outliers, or movement tracked between frames which is dissimilar to the background. In an attempt to correct for camera movement, the average velocity of the background is subtracted from all points, and velocities below threshold (near zero) are discarded. We assume the remaining outliers, depicted as red points, to be the hockey players.

The outlier selection is not always perfect and may count the stands as a foreground feature. To account for this problem, only outliers "on ice" can be valid. This distinction is made using an intensity mask introduced before background subtraction. A three step process is used to remove outlying, red, points from outside the ice surface.

Fig. 2. Results of optical flow, Homography and RANSAC on features, where the red points are the outliers discovered by homography using RANSAC. The green points with displacement vectors and warped image overlay are a result of the dual-stage optical flow algorithm.

First, a binary threshold by intensity is performed to force the frame into a black and white image, as shown in Figure 3. This is done using OpenCV's function, `threshold`, to change the threshold of the frame by intensity. Next, the contours recognized are enveloped by a convex hull, which is shown in Figure 4. OpenCV provides the `drawContours` that draws these contours and fills them in to remove the black images of the players, referees, and some parts of the boards.

A blur is applied to the contours to feather the mask and soften the edges, as shown in Figure 5. This helps to prevent players from having their upper bodies not taken into account because they are close to the boards.

Finally, using the blurred mask, every red point is checked to make sure that they fall within a certain intensity threshold indicating it is on-ice.

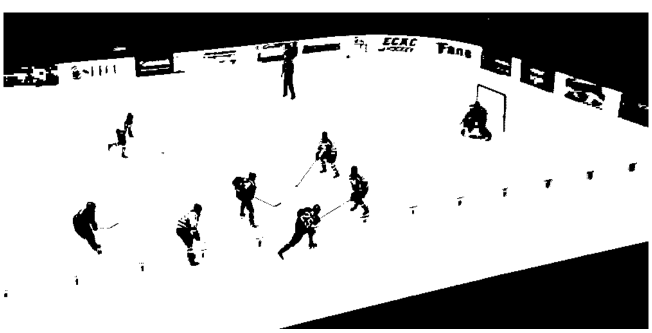

Fig. 3. Results applying a binary intensity threshold. The boards and players are black and the ice is white. The white ice begins to form the basis of the mask.

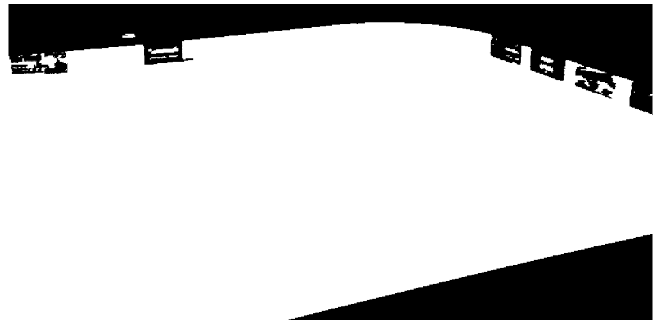

Fig. 4. Results of filling in found contours in the image. At this step, the mask has taken shape to include the ice and parts of the boards.

Fig. 5. Results of adding a blur to the mask.The blur softens the edges of the mask to prevent extraneous masking of the players. For example, if a player is near the boards, their head and chest may be above the boards and as a result be cut off.The softening of the mask helps to prevent on ice objects from being cut off.

In general, the remaining red points represent corners on players whose point velocities have been corrected. Distinct players are found using the mean-shift clustering algorithm. In general, the mean-shift algorithm locates dense areas of points. Without knowing the number of clusters ahead of time, we estimate the bandwidth using OpenCV's `estimateBandwidth`, a function which randomly selects 10 points and finds the average distance between them, called bandwidth. For each point, the clustering uses the bandwidth to find its neighboring points, then determines clusters by cross matching neighbors.

Once the point clusters are found, an average velocity is calculated for each cluster. The calculation identifies the displacement of the central cluster point across frames and divides by the duration of one video frame. Arrows are drawn using the OpenCV function `arrowedLine` at all points in the video to represent the velocity. The magnitude and direction of the arrowed line is drawn corresponding to the velocity measured for the cluster. In addition, a textual representation of velocity, measured in pixels per second, of each cluster is shown once the video is paused, allowing users to read the velocity measurement easily and without cluttering the screen.

Subsequent frames are processed using the same algorithm, where the current frame becomes the reference frame.

5. APPLICATIONS

The hockey speed program has many potential use cases, as well as a high potential to be used beyond the original scope of the program. Most notably, the main application of this program is to be used at a hockey game by users with smartphone quality video and tracking. Currently, the program has only been tested on the broadcast video from a Clarkson women's hockey game, so the capabilities of the system are unknown on mobile platforms. One caveat to the program is that poor player tracking is needed due to the way that motion in the background and motion in the foreground is detected. So, if a user perfectly tracks a player in the same position in frame for every frame, the program will be unable to determine the player's speed because it will be unable to separate the player from the background. Professional videographers and camera operators, such as those who manage the cameras during professional sporting events have the proficiency and equipment to precisely track players. For a typical smartphone user, stable tracking is an unlikely event.

Once the program is able to perform on mobile platforms, the possible use cases for the program widens significantly. The program should track and give velocity feedback about any moving objects that a user wishes to get information about. Spectating sporting events would be the most obvious application but there are also other applications. The program can be used as a lightweight and cheap training tool for coaches and trainers during practices and training sessions. One of the advantages of mobile platforms is that they can serve as edge nodes in an edge computing system. The data processing can be outsourced for centralized processing for more in-depth analysis. Such a system would greatly increase the computing power and abilities of the system. The ability to add more functionality to the program allows for a large amount of extendibility with the program.

6. RELATION TO PRIOR WORK

Optical flow has been implemented in the sports realm since the early 2000s. It has been used for general player tracking for sports like tennis, soccer, football, and ballet [8, 9, 10]. Since players are hard to track, especially when they may only be 15 to 30 pixels tall, optical flow was used for motion tracking along with target particle filtering for tracking players [8, 9]. The particle filtering was applied to different regions where movement of players occurs. Recognizing where players should be in the image is something considered in the algorithm discussed in this paper, however, instead the ice is detected by forcing the frame to black and white, brightening, and blurring the image. The resulting white space forms a convex a contour around it and all outlying features are discarded.

Others more specifically use the Lucas-Kanade algorithm for optical flow to find good features for tracking. One uses good feature tracking to navigate a robot around an environment and the other traces and extracts facial images for a noisy background [11, 12]. Similar to the algorithm discussed in this paper, both use real-time video. However, the navigating robot also uses GPS to help with positioning and orientation, while the facial images are coming from a static camera unlike the panning camera discussed in this paper.

Optical flow has been applied to hockey action recognition and player tracking. Researchers have used optical flow as part of an algorithm to detect violence, testing it to see if it recognizes hockey fights [13]. Others have used it to track players and tells users the action the player is taking such as skating forward or backward, however, this is performed on pre-recorded video [14]. Unlike the other hockey player detection and tracking applications the algorithm in this paper uses real-time video allowing for fans to engage with the action live. It also is self-contained and does not use an outside device such as a GPS for orientation. Finally, it applies Lucas-Kanade with a new way to identify the "field of play" by leveraging the white ice.

7. REFERENCES

[1] B. K. P. Horn and B. G. Schunck, "Determining Optical Flow," Artificial Intelligence, vol. 17, no. 1-3, pp. 185-203, 1981.

[2] S. Baker and I. Matthes, "Lucas-Kanade 20 Years On: A Unifying Framework," International Journal of Computer Vision, vol. 56, no. 3, pp. 221-255, 2004.

[3] OpenCV-Python (2018) [Online]. Available: https://pypi.org/project/opencv-python/

[4] D. Cournapeau. scikit-learn. (2018) [Online]. Available: https://pypi.org/project/scikit-learn/

[5] T. Oliphant. NumPy. 2018. [Online] Available: https://pypi.org/project/numpy/

[6] Open Source Computer Vision, “OpenCV-Python Tutorials,” Open Source Computer Vision, 2018. [Online]. Available: https://docs.opencv.org/master/d6/d00/tutorial_py_root.html

[7] M. A. Fischler and R. C. Bolles, “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography,” Communications of the ACM, vol. 24, no. 6, pp. 381-395, 1981.

[8] G. Zhu, C. Xu, W. Gao and Q. Huang, "Action Recognition in Broadcast Tennis Video Using Optical Flow and Support Vector Machine," in European Conference on Computer Vision, Graz, 2006.

[9] A. D. Narhare, G. V. Molke and R. Kanphade, "Motion Trajectory of Players using Optical Flow," in International Conference on Recent Trends in Information, Telecommunications, and Computing, Chandigarh, 2014.

[10] A. A. Efros, A. C. Berg, G. Mori and J. Malik, "Recognizing Action at a Distance," in IEEE International Conference on Computer Vision, Nice, 2003.

[11] Z. Vamossy, A. Toth and P. Hirschberg, "PAL Based Localization Using Pyramidal Lucas-Kanade Feature Tracker," Budapest Tech, Institute of Applied Informatics, Budapest, 2004.

[12] K.-S. Kim, D.-S. Jang and H.-I. Choi, "Real Time Face Tracking with Pyramidal Lucas-Kanade Feature Tracker," in International Conference on Computational Science and Its Applications, Kuala Lumpur, 2007.

[13] E. B. Nievas, O. D. Suarez, G. B. Garcia and R. Sukthankar, "Violence Detection in Video Using Computer Vision Techniques," in International Conference on Computer Analysis of Images and Patterns, Seville, 2011.

[14] W.-L. Lu, K. Okuma and J. J. Little, "Tracking and recognizing actions of multiple hockey players using the boosted particle filter," Image and Vision Computer, vol. 27, no. 1-2, pp. 189-205, 2009.