# Analyzing sports videos: Aligning and tracking the athlete to compare pole vault technique

Abigail Klein
Massachusetts Institute of Technology
kleinab@mit.edu

Nishanth Dikkala
Massachusetts Institute of Technology
ndikkala@mit.edu

## Abstract

*Development of computer vision techniques has greatly enhanced the analysis of sports videos. In this paper, we introduce a novel approach to analyze pole vault videos by overlaying two videos on top of each other in a semi-automated fashion, and display the athletes trajectory through the jump. To do this, we implement the computer vision techniques of background elimination and RANSAC, and introduce our own simple linear-time warping and tracking algorithms. In the resulting video, small differences in the athlete's two jumps are made clearer, leading to greater technical learning by the coach.*

## 1. Introduction

Pole vaulting is a specialized event in track and field in which athletes use a long, flexible pole to jump over a bar. In order to clear a high bar, athletes run full-speed down the runway and use the pole to convert their forward momentum to vertical height. The technique is extremely difficult to master, and slight variations in technique can result in huge variations in height. We develop a tool that supports coaches in understanding their athletes' vaults in order to give better corrections.

Most pole vault coaches watch videos of their athlete's jumps side-by-side and give corrections based on their observations. Oftentimes, jumps vary so slightly that it is difficult to differentiate better and worse jumps. We created an application that overlays two jumps on top of each other and tracks athlete through the video. This draws attention to differences in the athlete's position between one jump and the next, which will improve the coach's ability to find and correct technical errors. We will describe the restraints we place on our input in greater detail under assmuptions.

To achieve this goal, we address several challenges. First, the camera may shift laterally or rotationally over the course of a practice. Second, the coach may start the recording at an arbitrary time. This means that a particular phase of the vault (for example, when the athlete jumps) may not occur in the same section of the two videos. We need to align the videos spatially and temporally, respectively, to solve these problems. Finally, we wish to be able to distinguish the two trajectories in the overlayed video. To do this, we mark the trajectory the athlete takes in each jump and trace out the motion of their center of mass in the final overlayed video to help make the distinction clear. In order to solve all of the above problems, we need to extract the foreground and background. This extraction is a significant technical part of the paper upon which the application relies heavily.

### 1.1. Assumptions

The input to the system is two videos of the same pole vaulter, taken from a static camera with constant light. The camera position is largely constant with small rotational and translational movement only between the two camera frames. These shifts occur due to human error incurred during setup of the camera. We also assume that there are no prominent moving objects in the background and no occlu-

Figure 2. An example of two overlayed videos that were not temporally aligned. The image shows the athlete in two different phases of the jump, which is not useful to compare.

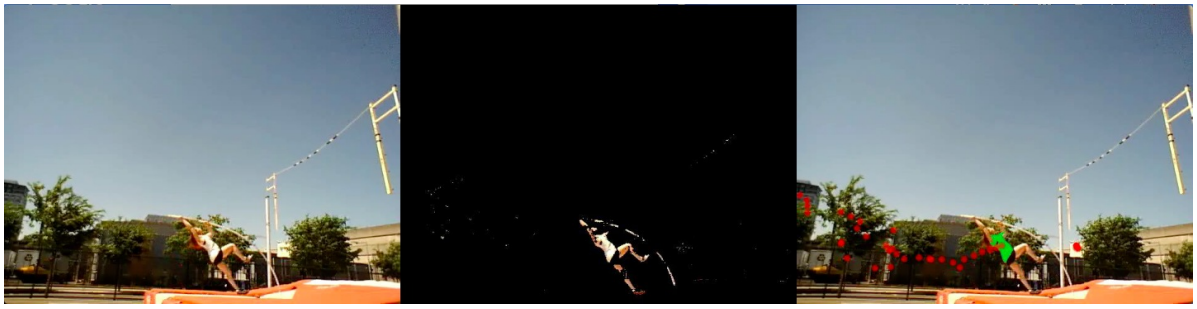

Figure 1. Three sample video frames from our results: the input (a), background-removed (b), and with the athletes trajectory displayed.

sion of the pole vaulter. The videos are short and the vault occupies a large fraction of the video. The videos are shot from the sideview of the athlete which mainly implies that the depth of the athlete in the camera frame of reference doesn't change leading to a roughly constant athlete size throughout the video.

### 1.2. Approach

First, we perform background elimination on the two input videos using the adaptive background elimination of Stauffer and Grimson [1]. We then spatially align the background using the RANSAC algorithm. We temporally align the foreground (consisting of the athlete and other moving objects) using our own linear time warping algorithm. We then track the athlete's movement from the foreground and plot a trajectory. Our approach to tracking is based on Patrick's report [**?**]. Finally we use alpha blending to average the spatially and temporally aligned videos.

## 2. Background

To perform background elimination we used the Adaptive Background Elimination algorithm of Stauffer et al [1]. Their technique also gives a way of tracking objects in videos by treating each connected component in the foreground as a moving object. We will elaborate on this technique in Section 3.

To spatially align the videos we need Fischler et als RANSAC algorithm [5] which requires SIFT feature computation as given by David Lowe [4]. This algorithm uses ideas from Harris corner detection [3] to detect good feature points and then creates SIFT descriptors for those points. Harris corner detection works by observing how the pixels in a window change when the window is shifted in all directions. If the change is significant, then we are probably at a corner. The SIFT algorithm, in contrast to Harriss algorithm, is robust to changes in scale and also works under presence of clutter or partial occlusion. Good feature points are defined as maxima and minima of the result of difference of Gaussians function applied in a scale space to a series of smoothed and resampled images. Low contrast candidate points and edge response points along an edge are discarded. Dominant orientations are assigned to localized keypoints.

These constitute the main technical references required to understand our project.

## 3. Methods

To align the two videos spatially, we must take care that the feature points chosen are not from a moving object. Hence we must select feature points from the background only. Also, it is easier to temporally align the foregrounds. For tracing the athlete's trajectory, again, we use a method which requires the foreground of the video. Hence, as a basis for all these tasks we perform background elimination first.

### 3.1. Background Elimination

Stauffer and Grimson's [1] paper on adaptive background removal describes an algorithm to remove background from videos that is robust enough to be deployed in uncontrolled scenarios for real-time surveillance. We present our implementation of their algorithm in brief. They suggest in their paper that a good model of the background is a weighted mixture of Gaussians for each pixel. In practice, multiple surfaces often appear in the view frustum of a particular pixel and the lighting conditions change. Thus, multiple, adaptive Gaussians are necessary to model the background. Based on the persistence and variance of each of the Gaussians of the mixture, it is determined whether the pixel belongs to the foreground or not. The means ($\mu$) of the Gaussians represent distinct pixel values which occur repeatedly and yet do not represent a moving object. For example, a periodically flickering light or a flag in the wind don't represent motion and it is well captured with this model. Noise around the means is well modeled by Gaussians.

If a pixel value lies far away from all the Gaussians currently in the mixture, it suggests one of two things

1. At that time instant, that pixel is part of the foreground

2. It could be evidence of a new Gaussian in the mixture

To allow for the possibility of case 2, we replace the least weighted Gaussian in the mixture by a new Gaussian centered at the current observed pixel value. In case the pixel value matches one of the Gaussians in the mixture, it leads to an increase in the weight of that Gaussian. This captures the guiding principles of the algorithm. Now, we describe the math.

The method contains two significant parameters, $\alpha$ the learning rate and $T$ the fraction of the pixels to be counted as a background. If $X_t$ represents the value of a pixel at time $t$, then the probability of observing $X_t$ is modelled as

$$P(X_t) = \sum_{i=1}^{k} \omega_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t}) \tag{1}$$

where $K$ is the number of distributions (between 3 and 5), $\omega$ represent the weights assigned and $\mu$ the means. $\Sigma$ represents the covariance matrix which for computational reasons is modeled as

$$\Sigma_{k,t} = \sigma_k^2 I \tag{2}$$

This assumes the red, green and blue pixel values to be independent and have the same variances. To update the distributions, we use an online $K$-means approximation. For every new pixel value, its deviation from the means of the Gaussians is computed. If this deviation is less than 2.5 times the standard deviation, the pixel value is said to match. The prior weights of the $K$ distributions are updates as follows:

$$\omega_{k,t} = (1 - \alpha)\omega_{k,t-1} + \alpha(M_{k,t}) \tag{3}$$

where $M_{k,t}$ is 1 for the distribution which matched and 0 for the rest. After this update, we renormalize the weights. The $\mu$ and $\sigma$ parameters remain the same for unmatched distributions. For the matched distribution,

$$\mu_t = (1 - \rho)\mu_{t-1} + \rho X_t \tag{4}$$

$$\sigma_t^2 = (1 - \rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T(X_t - \mu_t) \tag{5}$$

where

$$\rho = \alpha\eta(X_t|\mu_k, \sigma_k) \tag{6}$$

Under this model, if an object is stationary just long enough to become part of the background and then it moves, the distribution describing the previous background still exists with the same $\mu$ and $\sigma^2$ , but a lower $\omega$ and will be quickly re-incorporated into the background.

To decide what pixels represent the background, the algorithm maintains an ordered, open-ended list with most probable background distributions on the top and less probable ones gravitating to the bottom and eventually being replaced by new ones. (when a pixel isn't part of background

Figure 3. An example frame from the video with background eliminated. Notice that all the body parts were captured as foreground by the extraction algorithm.

it's value changes leading to Gaussian mixture modelling of its intensity profile over time). The distributions are ordered on the $\omega_k/\sigma_{k,t}$ value. For each pixel, the first $B$ distributions are chosen as the background where

$$B = argmin_b \left( \sum_{k=1}^{b} \omega_k > T \right) \tag{7}$$

Any pixel value which doesn't match with these $B$ distributions is regarded as background. This completes a description of the background elimination.

### 3.2. Spatial Alignment

We compute a transformation matrix from the background images of the two videos to spatially align the frames. First, we extract the background from the computed foreground video. Then, we use the Scale-Invariant Feature Transform (SIFT) algorithm to identify feature points in the backgrounds and the corresponding matches as shown in Figure 4. Finally, we use the Random Sample Consenus (RANSAC) algorithm to compute a homography matrix. The outline of the RANSAC algorithm is as follows: Do in a loop:

1. Select 4 feature pairs at random.
2. Compute the Homography (H) using the selected feature pairs.
3. Compute the inliers, that is feature pairs such that $||p_i^{'}, Hp_i|| < \epsilon$.
4. Keep the largest set of inliers.
5. Re-compute least-squares homography H using all the inliers.

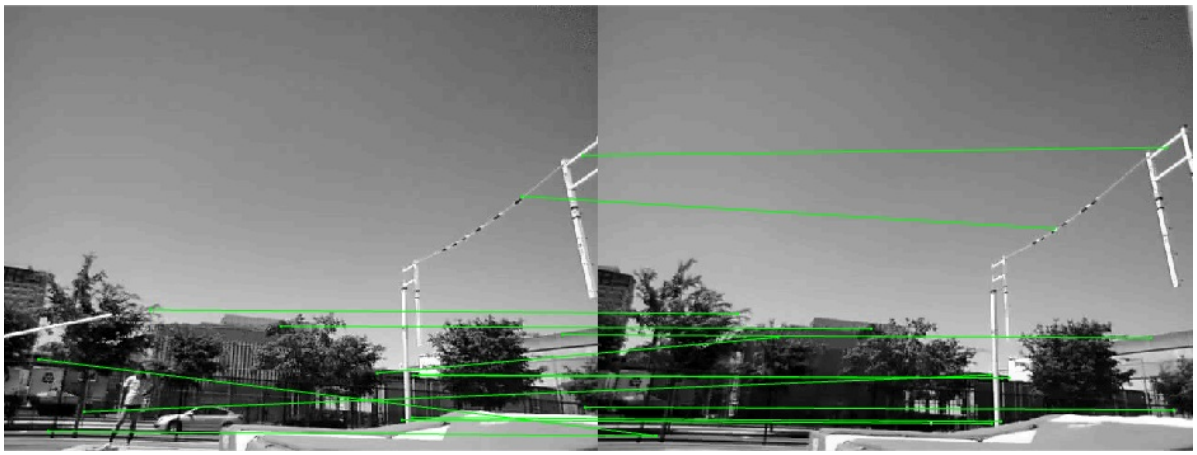

Figure 4. A sample of computed matches from the SIFT feature detector.

Because the camera is stationary, the background is constant throughout the video. Therefore we can compute a single transformation matrix to align all of the frames. The other reason we wish to align the background alone is that we wish to preserve changes in the athlete's pose between the two videos.

### 3.3. Temporal Alignment

Aligning two images has been well studied in computer vision. Aligning two videos introduces an extra layer of complexity: in addition to spatial $(x, y, z)$ alignment, videos must be aligned temporally $(t)$.

We introduce a linear-time warping algorithm to temporally align the videos. Because the frame rates of the two videos are the same and the athletes motion in each video is at full speed, the time shift is constant. We first create a binary mask of the foreground video by thresholding each frame of the foreground image at the value $\frac{0.5M}{255}$ where $M$ is the maximum pixel value occurring in the foreground image to get a binary image of 0s and 1s. For each frame in the first videos foreground, we compute the pixel difference between every frame in the second videos foreground. The minimum pixel difference is said to be the best match, and the number of frames shifted is saved. Once we have computed all of the best matches, we take their mode–i.e., the most common frame shift. We use this value to shift the second video to align with the first. Temporally aligning the foreground makes our application robust to two videos having different backgrounds.

### 3.4. Tracking and tracing the trajectory

We used a simplified tracking algorithm for tracing the athlete's trajectory based on the tracking technique suggested by Stauffer et al. [1]. First, the foreground was extracted for every frame, then the foreground image was converted into a binary image as described in the previous section. Then we computed the connected components of the foreground using the binary image [6]. Each connected component represents an independently moving object. We tracked the centre of mass (approximated by the centroid) of the largest connected component. The underlying assumption here was that for most frames, the largest connected component was the athlete. This proved to be a reasonable assumption as all pole vault videos primarily have two independently moving objects, namely, the pole and the vaulter. This assumption breaks when the background has prominent moving objects as well, but in videos which focus on the athlete, the area occupied the objects moving in the background is significantly less than that occupied by the athlete. Hence, for the videos we were interested in, the above algorithm produced satisfactory results.

We considered the idea of using continuity of the athlete's position to supplement the tracking algorithm, but in most of the videos, the athlete was absent in the initial few frames. This would have led to the algorithm tracking the wrong connected component and hence we trade-off slight noise in the initial frames of the video for correct tracking.

Also note that finding the connected components in the foreground allows us to compute not only the position of the moving object but also the size, moments and other shape information. These characteristics can be useful for later processing and enable greater technical learning.

### 3.5. Alpha-Blending to Overlay the videos

We perform $\alpha$-blending to overlay the two tracking videos and produce the final output. First, we use the temporal shift calculated during temporal alignment to determine which frames from the two videos we should overlay. Then, we align these two frames using the homography matrix computed during spatial alignment. We use an $\alpha$ value

Figure 5. Video frame showing the athlete's trajectory as a series of circles denoting where the athlete's center of mass was at each frame of the video. The athlete is colored green to denote her location.

of 0.5 to average the two frames.

## 4. Results

The adaptive background elimination algorithm is highly sensitive to the values of the parameters used but when their values are tuned, gives impressive results as shown in Figure 3. We used the number of Gaussians in the mixture for each pixel, $K = 5$, the learning rate $\alpha = 0.01$, the initial high standard deviation of a newly introduced Gaussian $HI = 0.12$ and the threshold indicating the minimum fraction of background pixels $T = 0.73$. We tested across different videos and the results were satisfactory in most frames of all videos. In some cases when the foreground is occluded by a part of the background, the algorithm ceased to detect some portions of the foreground. This happened, for example, when the bars occluded the pole vaulter during the cross-over.

The RANSAC algorithm performs relatively well to align the videos spatially. It misaligns the frames when the backgrounds had large differences, such as a crossbar being placed higher in one video than the next. In this case, SIFT correctly identifies feature points, but incorrectly identifies which feature points matched. On most of the videos that we tested, however, spatial alignment is satisfactory.

Temporal alignment produces good results for videos which satisfy the assumptions we make. Because the amount we shift by is the most common frame shift for each frame, if the input videos have larger portions of similar frames that are not related to the vault sequence (such as background), our temporal alignment algorithm might fail. Hence, the assumption that our videos are short and a major portion of them is the jump itself.

The tracking algorithm produces slightly noisy results for the first few frames because the athlete hasn't enetered the camera frame yet and as a result the largest connected component hasn't become evident. But as the video progresses the algorithm identifies the trajectory accurately enough to be able to tell the difference between two jumps. For the videos which satisfy the assumptions we make (minimal background noise), tracking produces a trajectory of the athlete's approximate center of mass (Figure 5).

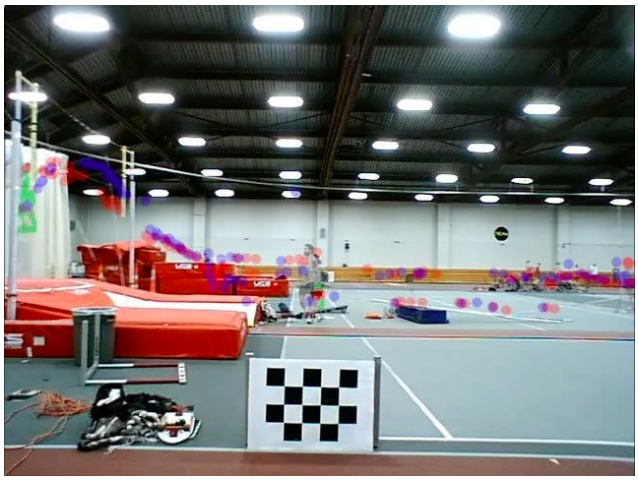

Figure 6. Video frame of the output of the system. Two videos are averaged and the trajectory of the athlete in each video is a different color (red/blue). It is clear from the output that that athlete is higher up in one jump than the other.

The entire system was found to be robust to changes in light (videos taken outdoors and videos taken indoors both produced good results).

## 5. Conclusions and Applications

In conclusion, we developed a tool which uses techniques from computer vision to achieve something useful in sport analysis. We implement background elimination, RANSAC, and our own linear-time warping and tracking algorithms and combine them to create our system. Notably, trajectory tracking can be used for more interesting applications which will further support the coachs technical analysis of the pole vault, such as computing the speed profile of the athlete.

## References


[1] Stauffer, C. and Grimson, W.E.L., Adaptive background mixture models for real-time tracking *Computer Vision and Pattern Recognition 1999(CVPR99)*, Colordo Springs, June 1999.

[2] Cen Rao, Alexei Gritai, Mubarak Shah, and Tanveer Fathima Syeda-Mahmood. View-invariant Alignment

and Matching of Video Sequences. *International Conference on Computer Vision (ICCV)*, page 939-945. IEEE Computer Society, 2003.

[3] C. Harris and M. Stephens. A Combined Corner and Edge Detector, in *Proceedings of the 4th Alvey Vision Conference* , pp. 147-151, 1988 .

[4] Lowe, David G. Object recognition from local scale-invariant features. *Proceedings of the International Conference on Computer Vision* 2. pp. 11501157, 1999.

[5] M. A. Fischler, R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Comm. of the ACM*, Vol 24, pp 381-395,1981.

[6] B. K. P. Horn. *Robot Vision*, pp. 66-69, 299-333. The MIT Press, 1986.

[7] Barragán, P. Moving Past Qualitative Coaching. 6.869 Final Report. CSAIL, Massachusetts Institute of Technology, 2009.