

Performing an 'Athletic Movement Assessment' for Sports Jump Using State of the Art Video Analysis and Heuristics Techniques like Body Detection and Displacement Assessment

Ali Sohani^a, Rafi Ullah^{b*}, Athaul Rai^c, Owais Karni^d

^{a,b,c,d} *Data Science Department, Cubix.co*

^a *Email: ali.sohani@cubix.co*

^b *Email: rafitullah.khan@cubixlabs.com*

^c *Email: athaul.raai@cubixlabs.com*

^d *Email: owais.karni@cubixlabs.com*

Abstract

This paper proposes a some novel and state of the art technique for analyzing the Athletic Movement (Vertical Jump) and feats by analyzing video frame by frame. Most common method to analyze "Athletic Movement" such as Jump and feats accomplished in them are either an observations made by an human expert / coach, or they are the values captured by measurement devices in the suit or wearables attached to the body of an athlete. Where former requires an access to the human expert, the later requires the special kind of a hardware / sensor that has capability to extract the body movement statistics with respect to time and space. Both methods are pretty accurate but due to their overhead in terms of necessity / dependence on 3rd party system or person. Not to mention along with the cost such methods come up with, they are often inaccessible in situations where one's just home practicing or when an athlete is just trying out something in own backyard or Gym (personal zones). Our target was here to reduce those dependencies and create such heuristics and algorithms that can help an individual athlete to assess the feats like Jump, Run, and Leap, without using any 3rd party systems, and be able to approximate the feats and compare them with the existing ones using only the cellphone device in their pocket. This paper focused on Jump sport. The system processed video frame by frame and Applying Histogram Of Oriented Gradient Technique to find Human in Frame and then track human from initial to last and we are capable now to calculate pixel distance covered by human in Jump. We used some values like human height to find physical distance covered, Frame Per Frame (FPS) of video, Markers on screen of mobile while recording videos.

* Corresponding author.

To validate the algorithm results, a number of experiments were performed and then compare with the actual vertical jump height and derive a statistical relation between the proposed methodology and the traditional techniques. Proposed technique can also be used for calculating different statistics of sport person.

Keywords: Vertical Jump Height; Sensor-less measurement; Video Analysis; Athletic Movement Assessment; Histogram of Oriented Gradient.

1. Introduction

The main objective of our research work is to analyzing athletic movement such as jump and feats accurately. Most common techniques and methods to analyze athletic movement and sports related activities requires access to human expert or some hardware (sensors) that have the capabilities of extracting body movement statistics with respect to time and space. Both of these methods are pretty accurate but due to their overhead and cost we are trying to eliminate them. But the most attractive reason of our work is to facilitate those whose wants to analyze their sport activities without using third party such as coach or sensors. We have tried to automate the traditional techniques [13] for finding vertical jump height such wall mounted, vertec, vertical jump mat, just jump mat, chronoJump, laser operated optoelectronic device, wearable system's and Force plate as discussed above. Using these traditional techniques involve the human judgment that may be erroneous and wearable techniques on the other hands is accuracy factor (mainly dependent on the quality of wearable sensors) but overhead for sport person.

We have used video processing technique including human detection state of the art algorithms Histogram Of Oriented Gradient, video meta data, Heuristics and simple mathematical operations for finding the vertical jump height of person using recorded video. We have apply human detection algorithm on every frame and keep record of the position of human body in frame, and then find the frame with highest position in video frame. Then we simply find the difference of two positions. This will be the pixel distance covered I-e vertical jump height. In order to convert pixel distance to actual distance we must have some assistance. For such assistance we used the height of jumper in video. If we the actual height of person, we use relative calculation to convert equivalent pixel distance into actual distance/jump height. The accuracy greatly dependent on the human detection process, if human is correctly detected and human detection Algorithm gives us perfect coordinate of human, then calculation will be fine and less accurate otherwise.

Rest of the paper include Related works, OpenCV, Histogram of Oriented Gradient, Traditional Technique for finding jump, Proposed methodology, Results, Conclusion and future work.

2. Related work

Reference [1] describes the working on same by using smart phone application that uses accelerometer sensors data for finding the vertical jump flight time. The sensor device is used on the sport person body and then collect the data and compared with smart phone data in order to validate the algorithm.

Plux is a body wearable sensors for such sports related activities to measure the sport statistics such jump flight

time etc. A wireless toolkit designed to collect and analyze highly reliable bio-signal data and some sort of similar toolkits for different purposes. This can be used for research purposes or educational purposes or may be for medical purposes [14].

Technical describe in [8] uses wearable (sensors) that send data to mobile device for analysis. These sensors are mounted on shoes, shirts and directly attached to human body.

The following references [19,20,21,22,23] are some of the software or techniques already available and in practice by the trainer and sports coaches. Almost all of these techniques require human intervention.

3. OPEN-CV

Gary Bradsky in 1999 at Intel for the very first time started work on OpenCV and in 2000 it release for the first time. **Vadim Pisarevsky** joined Gary Bradsky to manage Intel's Russian software OpenCV team. In 2005, OpenCV was used on Stanley, the vehicle that won the 2005 DARPA Grand Challenge. Later, its active development continued under the support of Willow Garage with Gary Bradsky and Vadim Pisarevsky leading the project. Now OpenCV support many algorithms related to Computer Vision and Machine learning and this expanding day by day. OpenCV Support a variety of computer programming languages including C++, Java, Python, C# etc., and available on Windows, Linux, OS X, iOS and Android platforms.

Interfaces for high-speed GPU operations based on CUDA and OpenCL are also under active development. We have used OpenCV-python library for our research work.

Python, a general purpose programming language started by **Guido van Rossum** that became very popular very quickly, mainly because of its simplicity and code readability. It enables the programmer to express ideas in fewer lines of code without reducing readability as compared to other languages.

Python is slower as compared to languages like C/C++. That said, Python can be easily extended with C/C++, which allows us to write computationally intensive code in C/C++ and create Python wrappers that can be used as Python modules. This gives us two advantages: first, the code is as fast as the original C/C++ code (since it is the actual C++ code working in background) and second, it easier to code in Python than C/C++. OpenCV-Python is a Python wrapper for the original OpenCV C++ implementation.

OpenCV-Python makes use of **Numpy**, which is a highly optimized library for numerical operations with a MATLAB-style syntax. All the OpenCV array structures are converted to and from Numpy arrays. This also makes it easier to integrate with other libraries that use Numpy such as SciPy and Matplotlib [5].

4. Histogram Of Oriented Gradient

Human detection in video/images is a challenging task. The task complexity is due different postures, actions, cluttered background, different variations and appearance of human in images. For such complex task we use HOG (Histogram of Oriented Gradient). HOG is a feature descriptor used in computer vision application for object detection in images. This technique simply counts the occurrences of gradient orientation in a localized

portion(s) of an image. There are also other methods for same purpose edge orientation histogram, shape contexts and SIFT (Scale Invariant Feature Transform descriptor), but HOG computer features descriptor on a dense of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy. OpenCV uses Support Vector Machine (SVM) for human detection or classification.



Figure 1: Figure showing HOG visualization

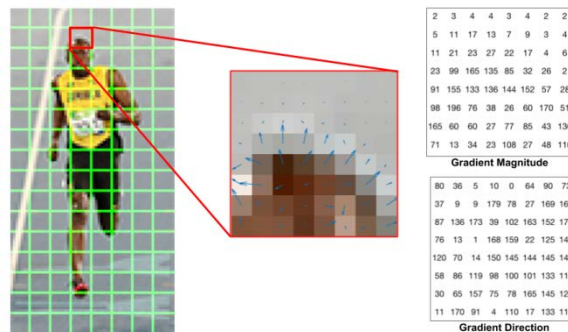


Figure 2: HOG applied on uniform grid of cells

5. Traditional Techniques for Finding Jump

There exist some simple techniques that can be and used for finding the height of jump. These methods are having their own advantages and disadvantages [9].

a. Vertec

The Vertec apparatus similar to Yardstick is commonly used for measurement of vertical jump ability. It is made of steel frame with having horizontal vanes that can be rotated through hands when the jumper jumps to some height. Each vane is in 1/2" increments (cm increments may be available), and the height of the vanes is adjustable from 6' to 12' to test elite athletes as well as beginners. The apparatus is very costly but you can use it inside as well outside.

b. *Wall Mounted*

The very old and original simple most technique used for measurement of jump height is a board having measurement on board mounted on wall. The jumper will jump near wall and the human expert will observe the readings. The technique is simple but often requires efforts to correctly observe reading. Technique has low cost but jumper must jump near wall having the board mounted on.

c. *ChronoJump*

ChronoJump, a multi-platform system for measurement, management and statistics of jump time events such as contact time, flight time etc, and other actions related to Jump sports. ChronoJump is an open hardware, free software complete system and this is distributed under the GPL license. You can have freely access to chronometer source code and program, and it is also possible to modify and redistribute the software. Because the source code is available, it is possible to verify the validity and reliability of the digital measuring instrument, making it appropriate for scientific investigations.

d. *Wearable Sensor*

Wearable devices / sensors are the modern latest technology used for mentioned purpose. There are applications available in Android and i-Phone that uses sensors and turn your mobile into jump height measuring device. Some sensors like Vert did same functions used in i-Phone. This device used the device's gyroscope and accelerometer to calculate the height of jump.

e. *Force plate*

For scientific purposes and research there exists devices like Force plates for measuring vertical jump height. These sensors not only tell you the jump height but also a lot of extra information available for interpretation of leg strength and power [15]. And many other technique such as Vertical Jump Mat, Just Jump Mat, Laser Operated Optoelectronic Device and Video system using equations of motions.

6. Proposed methodology

We have researched about the human detection and then tried to use these techniques for the sports related activities such as Jumping, leap, running etc. We have focused on finding the vertical jump height of person in video.

a. *Algorithm*

1..initial frame $\leftarrow 0$

2..final frame $\leftarrow 0$

3..height of person $\leftarrow h$

```
4..Pixel Distance = pix_dist = 0

5..vertical jump height = vjh = 0

6..//process video frame by frame

7..For frame in video frames

7.1..frame_y = y coordinate of frame

7.2..Apply HOG on frame using NMS

7.3../*use for first frame*/

7.4..if there is human in frame

7.4.1...if initial frame == 0:

7.4.1.1...//store y coordinate of frame

7.4.1.2...then initial frame ← frame and final frame = frame_y

7.4.1.3...continue

7.4.2...if final frame < frame_y:

7.4.2.1...then final frame = frame_y

7.4.3...else

7.4.3.1...discard frame

//use simple euclidean distance formula

/* we are only considering “y” point in vertical jump sport */

8..pix_dis = final frame – initial frame

//find the height of initial frame

// take only “y” point as it is vertical

9..if_y1 = initial frame lower left “y”
```

10..if_y2 = initial frame upper left “y”

11..height_video = if_y2 – if_y1

12.//convert pixel to real world distance

13.vjh = (h / height_video) * pix_dis

HOG is used along with SVM for detecting human in the scene, False positive reports I-e detecting more than one human as given in following Fig, we used Non Maxima Suppression (NMS).

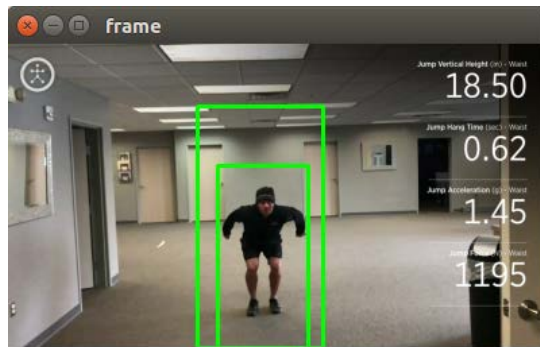


Figure 3: Person start jumping

b. Non Maxima Suppression

This technique is used in situation where your classifier detect more than one human in the image/frame of video, and actually there is one. NMS is used to remove other rectangles. Technque simply compute the areas of all bounding box one by one. Find the largest coordinate of rectangle. If there is sufficient overlap between between rectangle suppress the bounding box (rectangles), otherwise discard it.

After applying Non Maximum Suppression the smoothness in object detection occurs as shown in Fig.

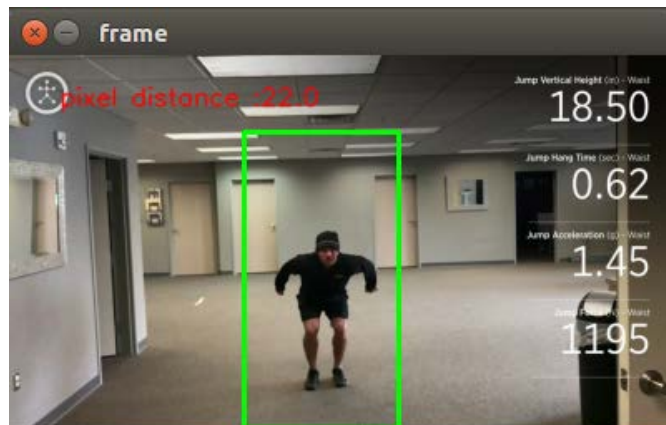


Figure 4: Person Detect After Applying NMS

Euclidean distance formula is given by

$$D(\text{point}_1, \text{point}_2) = \sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2} \quad (1)$$

But we are only interested in vertical distance, so we omit “x” point. So formula deduced to

$$D(\text{point}_1, \text{point}_2) = |y_2 - y_1| \quad (2)$$

where y_2 is the upper left or right coordinate or pixel value of the rectangle drawn on human body by Histogram of Oriented Gradient.

Conversion from pixel distance to actual distance using person actual height in meters or centimeters is computed using

$$\text{JumpHeight} = (h / \text{videoHeight}) * \text{pixelDistance} \quad (3)$$

where “h” is the height of jumper, “videoHeight” is the pixel distance of rectangle I-e height of rectangle, “pixelDistance” is the distance covered by jumper in video.

7. Results

We have successfully shown that this technique can be applied in sports related activities like jumping, running etc. Although there may arise different problems and situation where it may not works, but this can be used as an assistant or trainer for comparative analysis of sports persons different activities.

HOG detect human in video or image and gives us regions as well, where human is but the rectangle is little larger than person body (But this issue can be resolved by eliminating extra region as HOG gives this region in all most all cases) as shown in Fig below. We have used relative measures, I-e we supposed that rectangle by HOG is equal to the actual height of person in image. Then use this relative measure to convert pixel distance to actual distance.



Figure 5: Person start jumping in video

We here note down the height of this rectangle (subtracting lower “y” coordinate from the upper “y” coordinate) for later conversion of pixel distance to actual distance covered using person actual height. We keep track of frames until highest position amongst all frames in video. The highest frame is noted as shown in the Fig below.



Figure 6: Person at peak position while jumping

Table 1: Results of proposed system

Wall Mounted Results	Proposed Results	System % Accuracy	Comment
0.2286	0.18	78.74015748	
0.3556	0.23	64.67941507	
0.381	0.26	68.24146982	
0.3302	0.32	96.91096305	shoulders lifted and bend knees
0.4064	0.21	51.67322835	
0.3048	0.23	75.45931759	
0.4064	0.21	51.67322835	
0.3048	0.19	62.33595801	
0.2794	0.22	78.74015748	
0.3175	0.23	72.44094488	
0.6096	0.27	44.29133858	
0.381	0.25	65.6167979	
0.9144	0.3	32.80839895	bend knees
0.3048	0.22	72.17847769	
0.3048	0.23	75.45931759	
0.3048	0.14	45.93175853	
0.2794	0.17	60.84466714	
0.8128	0.21	25.83661417	bend knees
0.3302	0.18	54.51241672	
0.762	0.21	27.55905512	bend knees
0.4064	0.24	59.05511811	bend knees
0.2032	0.29	70.06896552	
0.1524	0.19	80.21052632	
0.2032	0.15	73.81889764	
0.2286	0.15	65.6167979	
0.7874	0.21	26.67005334	bend knees
0.2032	0.15	73.81889764	

Now we have actually the x, y points of two different positions that are initial position and final position. Now we simply subtract the “y” of initial frame from the “y” of final frame. This will gives us pixel distance. This pixel distance is converted to actual distance covered using person height. The person in video has height of 168 cm (66.50 inches) the proposed algorithm gives result of 13.513859548 inches Jump and in reality the jump was 1.10 feet (13.2 inches). We have performed many test cases and conclude following results in table 1. Table contains the results from proposed system, original result, the person height. We have compare our results against the wall mounted technique discussed above in the paper. We have set our test bed, we setup wall mounted devices and then records videos in front of it. We have observed results from proposed system shown in second column in table 1 and results from wall mounted apparatus has been carefully observed by human expert shown in the first column of Table 1. The accuracy has been shown in column 3 of table 1. The comment column in table 1 show what are the shape of human while jumping and recording videos. In some videos human bend their knees and take hands higher than body in some videos. Human actions likes knee bending, hand movement etc greatly effect the results. Following are compiled results

Table 2: Compiled Results

Avg Accuracy Calculated	74.4543595
Avg Accuracy without Knee bend	73.80775206
accuracy with knee bend	73.69861341
Avg. Accuracy	73.01

Proposed system has been tested outside environment without using wall mount on videos such as shown in Fig 5 and Fig 6 and the results are acceptable in all cases, but require improvement.



Figure 7: Person standing along with wall mounted apparatus

The person in frame in Fig 7 has height of 5 feet 9 inch (175.26 cm). This is the initial frame where person is going to start jump. Frame detected by Histogram of Oriented Gradient has been saved for later use.



Figure 8: Person extreme position at height

When person start jumping the initial frame is stored and then we keep track up to last frame the frame in which person is at highest point during his jump. We find that point by analyzing, when person position start decreasing, it means the person reach highest point. Now we have following information for calculating person jump height

- The initial frame x,y points
- The final frame x,y points
- The height of personal

We calculate the pixel distance using initial frame and final frame x, y position and then convert pixel distance to actual physical distance by using equation (3). There is one another issue with HOG when applied on video frames having noisy background or cluttered background. The bounded rectangle rectangle coordinates over shoot as shown in following figure.



Figure 9: Over shot frame during jump

This problem is removed by following technique

Algorithm remove over shoot frames:

```
1...frames [ ] ← HOG()
2...first_frame ← frames[0]
3...good_frames ← [ ]
4...for frames i←1 to length(frames)
5.....if avg(first_frame – frames[i]) <= mean(frames)
6.....first_frame ← frames[i]
7.....append frame[i] to good_frames
8...return good_frames
```

8. Conclusion

We have successfully implement the our proposed methodology on videos. The initial frame is noted, and then keep track of frames until highest position of person during jump. Then vertical jump height is calculated by subtracting the initial frame from final frame position in image. And then this pixel distance is converted to actual distance using the actual height of person jumping in video. And the results were tested against the actual values. The results seems perfect using video only I-e sensor less techniques. The problems arise are successfully overcame. One of the such techniques are NMS (Non Maximum Suppression) used. When HOG + SVM report more than one human bodies and actually there are exactly one human in the scene, NMS remove others. The technique mainly depend upon the quality of video, if this technique is applied in clear environment, results becomes more realistic. This technique can be made better using other techniques like CNN, Tracker, HOG and SVM.

Although this technique has accuracy less as compared to wearable sensors, or other techniques used in literature. But we have used only static mobile or normal camera video. This can used as a application in hand held devices. This technique has been shown work good and acceptable in indoor and outdoor environment as well.

9. Future work

Histogram of Gradient algorithm when applied on very low resolution video or scene where person is jumping in noisy or sparse background, HOG + SVM raise false positive results, sometimes it missed human in the scene, sometimes it detect other objects as human like sign, pictures in the scene, sometimes the rectangle is too small or too large, which cause measurements very in accurate. In future we will some robust techniques for

these mentioned issues. We will use trackers + HOG + SVM for accurate measurements along with some heuristics. We can also try advance machine learning techniques such Convolution Neural Network, but this will be slow. We are trying to do these calculations in real time in mobile device on live video streaming.

Acknowledgment

Thanks to my team, who has worked with me to follow the vision and lead. I am very thankful for their respect and hard-work. Also thanks to the organization, that gave us an opportunity to work on most pragmatic-level industry-grade problems and Research in the field of our passion (Data-Science).

References

- [1] Pires, Ivan Miguel, Nuno M. Garcia, and Maria Cristina Canavarro Teixeira. "Calculation of Jump Flight Time using a Mobile Device." In HEALTHINF, pp. 293-303. 2015.
- [2] Moir, Gavin L. "Three different methods of calculating vertical jump height from force platform data in men and women." *Measurement in Physical Education and Exercise Science* 12, no. 4 (2008): 207-218.
- [3] Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 886-893. IEEE, 2005.
- [4] Tsai, Grace. "Histogram of oriented gradients." *University of Michigan* 1, no. 1 (2010): 1-17.
- [5] Qiu, Shaohua, Gongjian Wen, Zhipeng Deng, Jia Liu, and Yaxiang Fan. "Accurate non-maximum suppression for object detection in high-resolution remote sensing images." *Remote Sensing Letters* 9, no. 3 (2018): 238-247.
- [6] Fan, Y., G. Wen, and S. Qiu. "An Accurate Framework for Arbitrary View Pedestrian Detection in Images." In *Journal of Physics: Conference Series*, vol. 960, no. 1, p. 012022. IOP Publishing, 2018.
- [7] Zhao, Chen, Zhao Yuqing, Yang Luqiang, Zhou Qiao, Gao Yanyu, Liu Cunrui, Wang Zihui, and Shi Ling. "Image Processing Techniques Based on OpenCV Spray Equipment." *Journal of Agricultural Mechanization Research* 6 (2018): 037.
- [8] George, Rana Praful, and V. Prakash. "Real-Time Human Detection and Tracking Using Quadcopter." In *Intelligent Embedded Systems*, pp. 301-312. Springer, Singapore, 2018.
- [9] Zhang, Shanshan, Rodrigo Benenson, Mohamed Omran, Jan Hosang, and Bernt Schiele. "Towards reaching human performance in pedestrian detection." *IEEE transactions on pattern analysis and machine intelligence* 40, no. 4 (2018): 973-986.
- [10] Li, Qingliang, Lili Xu, Pengliang Zheng, and Fei He. "A Local Neighborhood Constraint Method for SIFT Features Matching." In *Recent Developments in Data Science and Business Analytics*, pp. 313-320. Springer, Cham, 2018.
- [11] Tareen, Shaharyar Ahmed Khan, and Zahra Saleem. "A Comparative Analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK." (2018).
- [12] He, Tao, Yong Wei, Zhijun Liu, Guorong Qing, and Defen Zhang. "Content based image retrieval method based on SIFT feature." In *Intelligent Transportation, Big Data & Smart City (ICITBS), 2018 International Conference on*, pp. 649-652. IEEE, 2018.
- [13] <http://www.topendsports.com/testing/equipment-verticaljump.htm>, last visited 9-Oct-2017.

- [14] <https://www.plux.info/index.php/en/> , last visited 9-Oct-2017.
- [15] Linthorne, Nicholas P., Analysis of standing vertical jumps using a force platform. *Am. J. Phys.*, Vol. 69, No. 11, November 2001.
- [16] https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_setup/py_intro/py_intro.html, last visited 9-Oct-2017.
- [17] <http://scikit-image.org/> ,last visited 6-Oct-2017, last visited 9-Oct-2017
- [18] <http://pyimagesearch.com>, last visited 9-Oct-2017, last visited 9-Oct-2017.
- [19] <http://redlineathletics.com/>, last visited 9-Oct-2017.
- [20] <https://www.whatsmyvertical.com/>, last visited 9-Oct-2017.
- [21] <http://www.kinovea.org/en/> , last visited 9-Oct-2017.
- [22] <https://www.siliconcoach.com/>, last visited 9-Oct-2017.
- [23] <http://www.sportsmotion.com/> , last visited 9-Oct-2017.
- [24] <https://www.peakperform.com/> , last visited 9-Oct-2017.
- [25] <http://www.sportsclipmaker.com/> , last visited 9-Oct-2017.