Egocentric Height Estimation

Jessica Finocchiaro
Florida Southern College
jfinocchiaro@mocs.flsouthern.edu

Aisha Urooj Khan
University of Central Florida
aishaurooj@gmail.com

Ali Borji
University of Central Florida
aborji@crcv.ucf.edu

1 Abstract

First-person, or egocentric, video has been increasing in popularity in recent years with an increase in available technologies. Egocentric video is different than exocentric (third-person) video in some distinguishable ways, one of which being that the camera wearer is generally not visible in the video frames.

Recent work has been done on action and object recognition in egocentric videos, as well as work on biometric extraction from first-person videos. Here, we propose a method of estimating the height of an egocentric camera without any calibration or reference points.

2 Introduction

Use of first-person video (FPV) has increased in recent years with products like Google Glass [1], GoPro [2], and Narrative Clip [3] becoming increasingly affordable. Work has been done on biometric feature extraction in first-person video. However, to our knowledge, no one has done work on estimating height from egocentric frames.

To start, we recorded our own dataset of egocentric video recorded from three heights on a person- the waist, chest, and head. The first approach we took was to train a Support Vector Classifier (SVC) to determine if we could accurately determine which camera (waist, chest, or head) an image was taken from. After, we discretized the camera heights into smaller bins to see the accuracy we could attain with smaller height ranges, and we tested our SVC on different extracted features and layers of a pre-trained neural network.

To estimate a continuous height, we modelled a CNN based on Peleg's in [11]. We changed their network output to be regression-based, and used it to estimate height in shorter clips of video. We then

adjusted the network further to consider the spatial elements on the video. Instead of taking sparse optical flow as an input, the network was adjusted to take grayscale images as input, and we changed some of Poleg's activation functions as well.

Last, we combined the output of the temporal and spatial CNNs in a two-stream CNN to see if the combined networks could improve our results, like in [16].

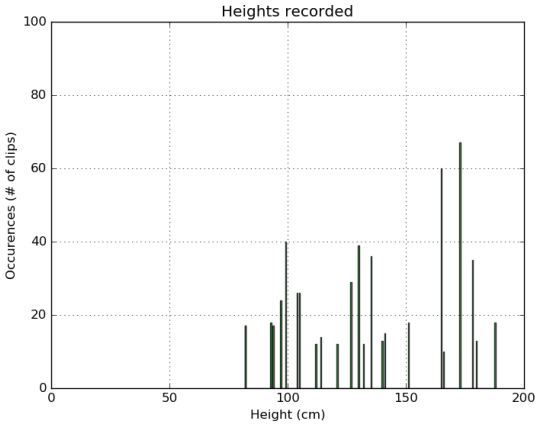


Figure 1: Distribution of video segments by height. Mean is 136.14 cm and standard deviation is 31.525 cm

3 Related Works

Egocentric video analysis has grown in recent years. It poses unique challenges since the camera wearer is not in the image and video does not have a stationary background, as the camera moves with the wearer. [17] demonstrates that Egocentric and Exocentric video pose unique challenges in video analysis and methods are more effective if analyzed differently. Works like [11], [13], [14], and [15] design CNNs to classify actions in egocentric videos. Works like [4], [6], [8], and [9] have also estimated height of objects or people in a video with a calibrated camera. Additionally, [12] uses saliency cues in egocentric video to predict objects of attention in egocentric frames. In [5], egocentric video is used to estimate the pose of the camera wearer without significant parts of the body being in the frame. Our problem, however, is unique since the person whose height we are trying to estimate is not in the image frame and using an uncalibrated camera makes the estimation more difficult.

4 Dataset

We recorded our own data for this project because there was no way of obtaining the knowledge to annotate a previously published dataset fitting our needs. We recorded ten people walking down a hallway with a Samsung Galaxy S4. Three different recordings were taken, at the head, chest, and waist, and height was recorded in centimeters. These were recorded once in a static setting, where there were no moving objects in the scene, and again in a dynamic setting, with other people walking through the hallway. Figure 1 shows the distribution of heights from which the video was recorded. It is important to note that we estimated the height of the camera, not the height of the person regardless of camera location.

5 Approach

5.1 Support Vector Classifier (SVC)

We trained a SVC to classify frames for three different ranges of height. To train the SVC, we used Linear and Polynomial kernels and performed cross-validation on the dataset. Different image features were extracted and vectorized to train the classifier for frames in the video. We also trained classifiers on the Raw Image Vectors as a comparison. Leave-One-Out testing was performed to see the results for each person's video with 3 bins.

5.1.1 3 bin classifier

The first bins we used to classify video was the location of the camera with respect to the recorder's body (ie. waist, chest, or head.) Feature extractors such as Histograms of Oriented Gradient, SIFT, Raw Image Vectors, Histograms of Oriented Optical Flow, GIST features, and various layers of pre-trained AlexNet [7] were used to train and test every 15th frame in a video. The results were verified with 3-fold cross-validation. After Histogram of Oriented Gradient, AlexNet layers consistently yielded the best results. In order to incorporate noise, we repeated our method with the videos recorded with dynamic backgrounds.

5.1.2 5 and 11 bin classifiers

Due to our small dataset, the test subjects whose results had the lowest accuracies were those heights were the extremes of the dataset. Because somebody who might be very tall might record their chest camera at a height close to the height of the head camera of a short person, we proceeded by training the SVC on smaller bins by centimeter height instead of camera location.

To start, we split data into 25 cm bins, yielding 5 bins with the range of our data. With a chance accuracy of 20%, results are shown in figure 2 for the different feature descriptors.

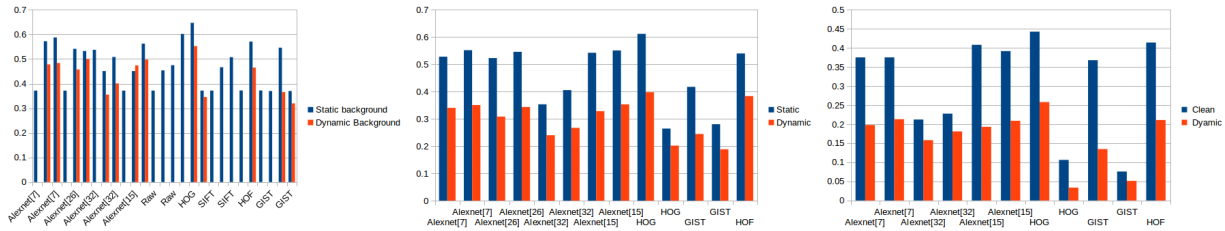
Similarly, we split the data into 10 cm bins, yielding 11 bins and repeated our training with a chance of 9%. Results are shown in figure 2.

5.2 Convolutional Neural Networks (CNNs)

Estimating height using a SVC has some drawbacks. For example, heights that are on the ends of each bin range might actually be closer in height to data in another bin. However, changing from classification to regression allows us to estimate height as a continuous measurement instead of in a discrete manner.

5.2.1 Temporal Network Architecture

Our temporal network is based heavily on Peleg et. al. in [11]. However, their group used the network to classify actions in clips of egocentric video into one of 7 or 14 categories. We modified the last layer of their network to have one output and used a Rectified Linear Unit as the activation



(a) 3 bins (b) 5 bins (c) 11 bins

Figure 2: Linear HOG consistently yielded the best results, followed by different layers of AlexNet and HOF. Linear kernel then polynomial shown for each descriptor.

function to estimate height. We normalized input values between 0 and 1 so the network generally returned a number between 0 and 1. We then converted the values back using the same scale to provide a height estimate.

Video segments were normalized to 15 frames per second. A sparse 32x32 optical flow vector was extracted from the frames and the x and y values were temporally concatenated, then the frames over 4 seconds were concatenated in the same manner, making the input to the network 32x32x120 optical flow vectors. Video collection started every 2 seconds so there was overlap in the videos. A height estimate was taken for every 4 second clip, and the video's estimate was given by taking the mean of estimates for each clip in a given video.

The network starts with a 3D convolutional layer with 30 kernels of size 17x17x20 with spatial stride of 2 and a temporal stride of 4, since flow vectors are concatenated on the temporal dimension. 3D pooling of size 2x2x13 is then applied with a stride of 2x2x13. Output is size 4x4x2, and 2D Convolution is the next hidden layer with 100 kernels of size 3x3. The output is 2x2, and max pooling is applied to make the output 1x1. Fully connected layers of size 400, 50, and 1 are added to yield one output. All activation functions are Rectified Linear Units (ReLU.)

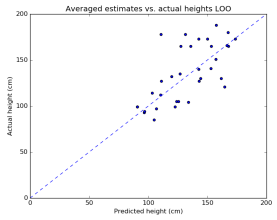
5.2.2 Spatial Network Architecture

The architecture of our spatial network is very similar to that of the temporal network, but instead of taking optical as the input to the network, pixel intensities of grayscale images are used. Because of that, the network input is 32x32x60. Additionally, instead of using ReLU activation functions, we used an Exponential Linear Unit and Batch Normalization. [11]

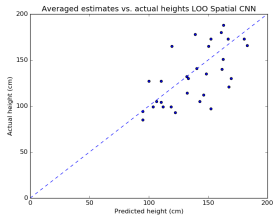
6 Results

6.1 SVM

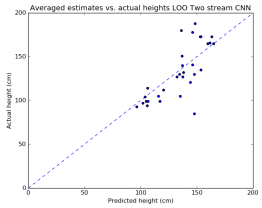
Multiple trials with the SVM were conducted across different bin sizes and training sets. While we compared the results using ev-



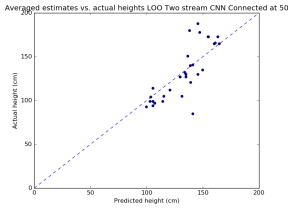
(a) Temporal network



(b) Spatial network.



(c) Two stream network connected at final output.



(d) Two stream network connected at 50 neuron output.

Figure 3: LOO testing results (estimate v. actual) on different CNNs

ery 15th frame for static and dynamic backgrounds, (figure 2) our main interest was finding the feature descriptor and kernel that were most successful in classifying the approximate height of the camera. Using a static background, Histograms of Oriented Gradient (HOG) with a linear kernel was most accurate in classifying the approximate height of a camera across all bin sizes. For 3 bins, 5 bins, and 11 bins, the HOG with a linear kernel classified with 64.69%, 61.15%, and 44.2% accuracies, respectively. See figure 2 for complete results.

6.2 Neural Networks

For each network, we performed Leave-One-Out (LOO) training by person on the dataset. Note that for each round of LOO training, there are actually three videos being tested since we test by person. There are usually 10-25 normalized video clips per video, and one estimate is yielded per segment. The estimate for the entire video is given by the average of the estimates for each clip.

6.2.1 Temporal Network

Using sparse optical flow as network input with LOO testing, the Mean Absolute Error (MAE) is 18.03 cm and Mean Squared Error (MSE) of 549.28. The r^2 value is 0.4557. A scatter plot of the estimates is shown in figure 3.

6.2.2 Spatial Network

With pixel intensities as network input, LOO testing yields a MAE value of 19.51 cm and MSE value 602.54. The Temporal Network alone, therefore outperforms the Spatial Network. However, a two-stream network outperforms either network by themselves. r^2 with the spatial network is 0.4267.

6.2.3 Two stream network

The two-stream network had LOO training performed twice. Once, the network was only trained on the final outputs of each layer. In this case, the MAE was 14.53 cm with a MSE value of 323.98 and r^2 value of 0.5773. By all three metrics, the two-stream

Table 1: Comparison of LOO training results

	MAE	MSE	r^2
Temporal	18.03	549.28	0.4557
Spatial	19.51	602.54	0.4267
Two stream- 1	14.53	323.98	0.5773
Two stream- 50	14.04	303.48	0.6505

network outperformed both the spatial and temporal networks.

We also performed LOO training on the two-stream network, but instead of concatenating the outputs of each network, we trained it the second time based on the outputs of the fully connected layer that has 50 neurons to see if this improved results. Here, we had MAE of 14.04 cm, MSE of 303.84, and an r^2 value of 0.6505. These results all improve on the network when merged on the final output, albeit only slightly when considering MAE.

A full comparison of results can be seen in table 1.

7 Conclusion

In this paper, we were able to extract important information about human vision. While it is obvious that people see the world differently at different heights, we were able to train an SVM on different features to learn how a computer most successfully perceives these differences in estimating height. Additionally, we were able to train Convolutional Neural Networks based on temporal features, spatial features, and a combination of the two to learn height estimation in egocentric video. To our knowledge, this is the first research done on height estimation and can be used to extract biometric information about a camera wearer. This further supports Peleg's claim in [10] that egocentric camera wearers should share with caution, because they might not as anonymous as they seem, despite rarely entering the frame.

8 Future Work

In future work, we plan to investigate different areas of biometric information extraction from egocentric video. We want to investigate gait signatures, similar to the work of Peleg et. al. in [10]. We are also interested in extracting the speed of an egocentric camera wearer and potentially training a network to diagnose different types of gait disorders from First-Person Video.

References

[1] Google glass. `https://www.google.com/glass/start/`.

[2] Gopro. `https://gopro.com/`.

[3] Narrative clip. `http://getnarrative.com/`.

[4] C. BenAbdelkader and Y. Yacoob. Statistical body height estimation from a single image. 2007.

[5] K. Grauman and H. Jiang. Seeing invisible poses: Estimate 3d body pose from egocentric video. 2016.

[6] I. Kispál and E. Jeges. Human height estimation using a calibrated camera. 2006.

[7] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. 2012.

[8] J. Ljungberg and J. Sönnerstam. Estimation of human height from surveillance camera footage- a reliability study. 2008.

[9] M. Momeni-K, S. Diamantas, F. Ruggiero, and B. Siciliano. Height estimation from a single camera view. 2010.

[10] S. Peleg and Y. Hoshen. An egocentric look at video photographer identity. 2015.

[11] S. Peleg, Y. Poleg, A. Ephrat, and C. Arora. Compact cnn for indexing egocentric videos. 2015.

[12] J. Rehg, A. Fathi, and J Hodgins. Social interactions: A first-person perspective. 2012.

[13] J. Rehg, Y. Li, and Z. Ye. Delving into egocentric actions. 2015.

[14] M.S. Ryoo, Y. Iwashita, A. Takamine, and R. Kurazume. First-person animal activity recognition from egocentric videos. 2014.

[15] M.S. Ryoo and L. Matthies. First person activity recognition: What are they doing to me? 2013.

[16] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. 2014.

[17] C. Tan, H. Goh, V. Chandrasekhar, L. Li, and J. Lim. Understanding the nature of first-person videos: Characterization and classification using low-level features. 2014.