Human Detection and Height Estimation

1st Daivat Bhatt
School of Engineering and Applied Science
daivat.b.btechi15@ahduni.edu.in

2nd Mohit Solanki
School of Engineering and Applied Science
mohit.s.btechi15@ahduni.edu.in

3rd Rahul Savariya
School of Engineering and Applied Science
rahul.s.btechi15@ahduni.edu.in

4th Neel Shukla
School of Engineering and Applied Science
neel.s.btechi15@ahduni.edu.in

5th Jeet Trivedi
School of Engineering and Applied Science
jeet.t.btechi15@ahduni.edu.in

***Abstract*—Our project aimed to tackle two essential problems in surveillance: human detection and height estimation. Both the aspects are necessary and important in identifying threats and are therefore problems that constantly require a solution. In this paper, we will go over our methods of human detection (performed using YOLO Algorithm) and the subsequent tracking of the human (using Kalman Filter), and the height estimation (attempted using least square fit).**

***Index Terms*—Human Detection, Height Estimation, Image Processing, Surveillance, Camera Calibration**

I. Introduction

Our project consisted of two parts - human detection and the height estimation. We intended to simulate a scenario observed in surveillance cameras where it was important to identify, track and then make a character-sketch of the identified person using the height. We settled into height as our soft biometric criteria for it was one of the primary and easily identifiable traits of humans. It also helps to broadly categorize people thus helping us to weed out non-matching suspects. We attempted to perform both the activities independently. We successfully implemented the YOLO algorithm implementation of human detection, whose details and results will be discussed in the later half. In the height estimation part, we had to change our perspective at attempting the problem. We initially had planned to use the inbuilt MATLAB functions and a checkerboard to obtain the camera parameters thus calibrating the camera. However, we discovered that the values that the function gave inaccurate results and therefore scrapped the idea. We then moved to manually marking head and foot points of a person, using the known height and then using mathematical functions to fit a curve into the data. This would help us gain a rough approximate of the focal length, tilt angle and the camera height.

A. Contributions

The work was divided, yet we knew that both the problems were eventually linked together. It was assigned to the members as follows:

- Daivat Bhatt : Literature Review (Height Estimation), Formulation and syntactic implementation of corresponding approach, Coding of the Height Estimation part
- Jeet Trivedi : Literature Review (Height Estimation), Coding of the Height Estimation part
- Neel Shukla : Formulation and syntactic implementation of corresponding approach, Coding of the Human Detection part
- Mohit Solanki : Literature Review (Human Detection), Formulation and syntactic implementation of corresponding approach
- Rahul Savariya : Literature Review (Human Detection), Formulation and syntactic implementation of corresponding approach

II. Related Work

Our attempts to tackle both the problems had contrasting results.

While the work on human detection was fairly simple and had been achieved by the time the Second Evaluation was due, the height estimation part was posing multiple problems.

Primarily, this was because we had been caught up in the checkerboard pattern method to obtain the camera parameters. After multiple attempts at trying to estimate the parameters (which were discussed in the previous report) and after a discussion with Professor Mehul, we decided to look for other options.

Regarding the human detection part, our approach was very clear that we should use YOLO algorithm. We had explored the fast-RCNN as well as the faster-RCNN framework but it was computationally heavier and hence we did not want to compromise on instantaneous output. The YOLO algorithm gives an accurate output when used on the surveillance camera on the first floor, GICT campus, Ahmedabad University.

III. Our Approach

After careful research, we found a mathematical method of finding out the parameters using the least square fit function. We could then use these parameters to find the heights of the same or the different subject in the frame. The human detection

part was, as discussed previously, attempted using the YOLO algorithm.

A. Human Detection

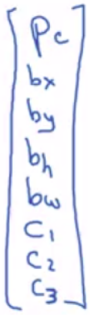
The YOLO algorithm consists of five components:
- Object Localization
- Object Detection using Sliding Windows
- Intersection over Union
- Non Maximal Suppression
- Anchor Boxes

The above components will be explained in the later half of this subsection.
The YOLO algorithm was applied on the test dataset by using pre-trained weights tiny-yolo-voc-1c.cfg.
The details about the convolutional neural network used are given below:

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

1) Object Localization: First objective in the video frame is how to represent an object when detected in a video frame. Assume for now that you would like to detect the persons: p1, p2, p3.

The image below shows how to represent an object when it is detected. It will be a 1x8 row vector Pc : whether human is there in the image bx : X co-ordinate of the bounding box by : Y co-ordinate of the bounding box bh : Fraction of Height of the bounding box w.r.t the grid cell bw : fraction of Width of the bounding box w.r.t the grid cell p1, p2, p3 : 0 or 1 (1 if detected)



2) Object Detection using Sliding Windows: It takes a grid cell in the image of a particular size and checks for the existence of an object in those grid cells and notes down the bx, by, bw, bh values.

3) Intersection over Union: Normally there is always a deviation between the ground truth bounding box and the evaluated bounding box. We use IOU to solve this issue.

$$\text{IOU} = \frac{\text{Area(Ground Truth} \bigcap \text{Evaluated)}}{\text{Area(Ground Truth} \bigcup \text{Evaluated)}}$$

Generally, IOU $\geq$ 0.5 is considered.

4) Non Maximal Suppression: The same object can be detected in multiple grid cells. What it does is take the bounding box with the highest accuracy of that object detected in the image will be taken into consideration. Thereby taking the most accurate detection.

5) Anchor Boxes: If the bounding boxes of the detection of the objects overlap, anchor boxes are used. Object Detection occurs normally and then IOU is calculated w.r.t. the anchor boxes. These are usually hand designed bounding boxes. And now the object localization matrix will be a 1x(m*8) matrix; m being the number of anchor boxes defined.

Tracking using Kalman Filter

A Kalman Filter is basically a tool that helps us to predict values. What our objective now was to essentially fill in the gaps between the detections that the YOLO algorithm was not able to do. The object should be tracked in each frame of the video. It is a 2 step process:
- Predict:

Here, we predict the new value called predicted value based upon the initial/previous value and then predict the uncertainty/error in our prediction according to the various measurement noise present in the system.
- Update:

Here, we take in account the actual measurement coming from the device and we call this as measured value. Here we calculate the difference between our predicted value and measured value and then decide which value to keep by calculating the Kalman Gain. We then calculate the new value and new error based on our decision made by Kalman Gain. These calculated values will finally be the predictions done by our Kalman Filter in iteration 1.
The output of the update step is again fed into the Predict State and the cycle continues until the error between our predicted and actual values converge to zero.

B. Height Estimation

The basic pinhole mode and its equation is given as follows:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = K \begin{bmatrix} R & T \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Fig. 1: Matrix Equation

The entire matrix except the coordinates part could be written in the following method:

$$\mathbf{P} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -c \\ 0 & 0 & 1 & 0 \end{bmatrix}$$
$$= \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f\cos\theta & -f\sin\theta & -fc\cos\theta \\ 0 & \sin\theta & \cos\theta & -c\sin\theta \end{bmatrix},$$

Fig. 2: Pinhole Matrix

Using these values and putting them in the actual equation where the conversion from the real world, we get:

$$\begin{bmatrix} \mathrm{x} \\ \mathrm{y} \\ \mathrm{w} \end{bmatrix} = \mathbf{P} \cdot \begin{bmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{bmatrix}$$
$$= \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f\cos\theta & -f\sin\theta & -fc\cos\theta \\ 0 & \sin\theta & \cos\theta & -c\sin\theta \end{bmatrix} \cdot \begin{bmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{bmatrix}$$
$$= \begin{bmatrix} f\mathrm{X} \\ f\cos\theta \cdot \mathrm{Y} - f\sin\theta \cdot \mathrm{Z} - fc\cos\theta \\ \sin\theta \cdot \mathrm{Y} + \cos\theta \cdot \mathrm{Z} - c\sin\theta \end{bmatrix},$$

As the human is obviously vertical to the ground, the y-coordinates can provide more information and therefore help in the eventual optimization of the values. Therefore, the y-coordinate equation could be written as follows:

$$\mathrm{y_f} = \frac{f\mathrm{Y_f} - f\tan\theta \cdot \mathrm{Z_f} - fc}{\tan\theta \cdot \mathrm{Y_f} + \mathrm{Z_f} - c\tan\theta},$$
$$\mathrm{y_h} = \frac{f\mathrm{Y_h} - f\tan\theta \cdot \mathrm{Z_h} - fc}{\tan\theta \cdot \mathrm{Y_h} + \mathrm{Z_h} - c\tan\theta}$$

where the y_h and y_f are the y-coordinates of the head and foot points.

Now, put $\mathrm{Y_f}$ as 0, and $\mathrm{Y_h}$ is the actual height of the human in question. Also, the Z co-ordinates will require an extra dimension and further measuring grids and are therefore ignored.

Upon simplification, we are left with the following:

$$\mathrm{y_h} = \frac{f\left(-c\tan^2\theta + \mathrm{Y_h} - c\right)\mathrm{y_f} + f^2\tan\theta \cdot \mathrm{Y_h}}{\tan\theta \cdot \mathrm{Y_h}\mathrm{y_f} + f\left(\tan^2\theta \cdot \mathrm{Y_h} - c\tan^2\theta - c\right)}$$

As we can see above that the equation has a non-linear form, its parameters can be found by fitting a curve and using non-linear regression.

After the parameters are found, the estimated height can be found as:

$$\hat{\mathrm{Y}}(\mathrm{y_f}, \mathrm{y_h}) = \frac{-\hat{f}\,\hat{c}\left(\tan^2\hat{\theta} + 1\right) \cdot (\mathrm{y_f} - \mathrm{y_h})}{\tan\hat{\theta} \cdot \mathrm{y_f}\mathrm{y_h} - \hat{f}\,\mathrm{y_f} + \hat{f}\tan^2\hat{\theta}\mathrm{y_h} - \hat{f}^2\tan\hat{\theta}}$$

IV. Experiments and Result

A. Human Detection and Tracking

The results of the human detection and tracking algorithm are shown below. We used a 10-minute time frame to observe different people walking in and out of the camera view, and these people were correspondingly tracked:

Fig. 3: Results of Human Detection algorithm

B. Height Estimation

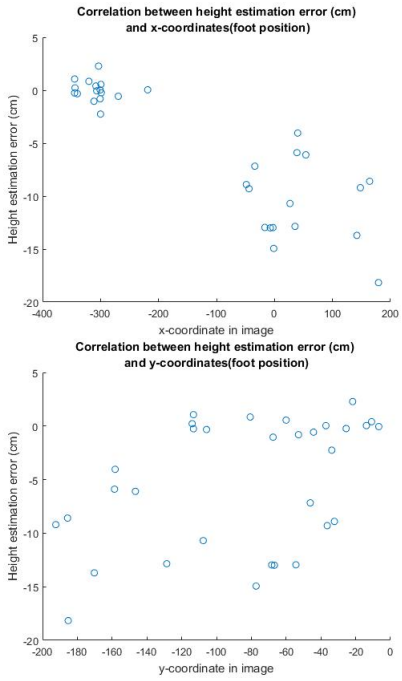
We were able to implement the mathematical equations, as well as fit a non-linear curve. However, the results were inaccurate by a considerable margin. This may be attributed to the fact that there were not a lot of samples available for there to be enough data-points. This of course would result in a curve that is not a good enough approximation of the camera parameters required.

The camera parameters obtained were:

$$f = 0.3556mm$$
$$\theta = -32.3367°$$
$$h = -2.1466m$$

θ is the tilt-angle measured from the right-angular position of a surveillance camera. The negative sign is merely for purposes of denoting its downward direction.
The height h is in meters, and is negative because we consider the camera to be the origin and therefore the ground is on the negative y-axis.

We attempted to find a correlation between the x-coordinate of the image and the y-coordinate of the image with that of the height measured of the subject. The results are given below. As we can make out, there is a distinct correlation between the both. As the person moves away from the center of the image (the origin in this case), the radial distortion begins affecting the height and therefore it will gradually decrease. When the person moves to and fro from the camera, the height follows a linear dependency shown in the second image.



The parameters were measured using one subject's known x coordinates, y coordinates and the height. After the a non-linear curve was fit, these values were used to estimate the height of the second subject. The height of the second subject was 172cm, while it was estimated at 162cm. Clearly, there is a considerable discrepancy.

V. Conclusion

The YOLO algorithm was, as you can see, successfully implemented using the same surveillance obtained from the campus.
However, the height estimation part was laden with errors which might have been solved if we had more subjects as well as more frames with discernable head and foot points. We were able to generate a good-enough estimate of the focal length and it matched with the actual focal length. But, it would have been desirable had there been more people and data points. This could be linked with the fact that we spent a major amount of time in calibrating the camera using te checkerboard pattern.

Acknowledgment

The authors would express their sincere gratitude to Dr. Mehul Raval for his influence on the course of this work and for his suggestions in the improvement of results. We would also like to acknowledge the assistance provided by Mr. Vandit Gajjar and Mr. Hiren Galiyawala.
We would also like to thank Mr. Rahul Bhatiji as well as the IT team at School of Engineering and Applied Science, Ahmedabad University in providing us the surveillance videos from cameras on campus.

References

[1] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. Conference on Computer Vision and Pattern Recognition (CVPR), 2005.
[2] Girshick, Ross et al. "Rich Feature Hierarchies For Accurate Object Detection And Semantic Segmentation." Arxiv.org. N. p., 2013. Web. 26 Sept. 2018.
[3] Tsai, R.Y., 1987: A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. IEEE Int. Journal Robotics and Automation, Vol. 3(4), pp. 323-344
[4] Heikkil, J. and Silven, O., 1997: A four-step camera calibration procedure with implicit image correction. CVPR97
[5] E. Jeges, I. Kispal and Z. Hornak, "Measuring human height using calibrated cameras," 2008 Conference on Human System Interactions, Krakow, 2008, pp. 755-760.doi: 10.1109/HSI.2008.4581536
[6] E. Jeges, I. Kispal, "Human height estimation using a calibrated camera", Oldweb.mit.bme.hu. N. p., 2018. Web. 15 Oct. 2018.
[7] Redmon, Joseph et al. "You Only Look Once: Unified, Real-Time Object Detection." Cv-foundation.org. N. p., 2016. Web. 15 Oct. 2018.
[8] Li, Shengzhe et al. "A Simplified Nonlinear Regression Method For Human Height Estimation In Video Surveillance." EURASIP Journal on Image and Video Processing 2015.1 (2015): n. pag. Web. 12 Nov. 2018.