3D Person Tracking in Retail Stores

Russell Kaplan and Michael Yu
Stanford University
450 Serra Mall, Stanford, CA 94305
{rjkaplan, myu3}@stanford.edu

Abstract

In this project, we sought track the movement of multiple people in 3D given security footage that is representative of what would be available in retail stores without modifying existing camera deployments. More specifically, this involves using a single, possibly fisheye-distorted view to track people in 3D space and model where they are in a room or store. This involved bridging various papers across the computer vision literature, looking at radial distortion resolution for images from an uncalibrated camera; calibration techniques from single-view metrology, affine approximation, and a simplified special-case mathematical model for object detections on a ground plane; and deep region-based convolutional networks for 2D person detection.

We define an approach that uses this single security camera view to track people on a ground plane, relying on some assumptions about the geometry of the space, but no additional hardware, giving our work an advantage over existing companies and processes that rely on more sophisticated sensors or sensor networks for person-tracking. The assumptions we make are realistic in the retail store context, and our quantitative results are compelling: significantly more accurate than WiFi-based tracking solutions that are already being deployed [9]. Our approach can be deployed in most retail spaces without any hardware modifications to existing security setups. We are excited by the real-world applicability of our work.

1. Introduction

As we see much of retail moving online from brick-and-mortar, opportunities to analyze consumer shopping behavior are rapidly growing and becoming commercialized. Recommendation engines, product positioning on webpages, and sales funnels are all relentlessly A/B tested and optimized to seduce the consumer into clicking Buy. Much as the abundance of data makes optimization and iteration easy in the e-commerce space, brick-and-mortar retail stores are looking to obtain and harness similar data to shelve products, auction shelf space, and strategically place discounts and information.

There are numerous current approaches being taken to obtaining real-time consumer data in retail stores. Many of these involve tracking peoples movement throughout the store, with Bluetooth or wifi tracking, or with photogates. All of these require significant hardware and deployment costs, which has hindered their scalability.

Computer vision is a promising tool to address this business need. As weve seen significant progress in the field as of late (notably from convolutional neural networks), cameras are an attractive option to track users in a store and obtain data on customer retail behavior. Cameras are a particularly attractive method to obtain this data because most retail stores already have the necessary hardware in place for the purposes of security.

In this project, we track the 3D positions of multiple people throughout a store in real-time, given a camera source and certain assumptions about the store layout (namely, that only one floor is visible to the camera, and that people's feet are visible in the image frames). We show a live 3D bounding box whose coordinates are relative to the world frame that tracks each person as they move, with error levels within 55cm in outdoor settings and 30cm indoors, much better than the existing WiFi- and Bluetooth-based tracking solutions [9]. To do this, we tie together numerous concepts from computer vision in a single approach - we design an entirely new approach for this business need by integrating various approaches. To each subproblem, we tried various options, picked the best, and identified optimizations for this particular case where applicable - for instance, we found that affine approximation of tiled floor grids worked far better than vanishing points given the configuration of many security cameras with respect to the floor.

In this paper, we beginning by examining the problem statement, and related work, both work that we read to gain background knowledge, and also that we read to learn from and build on their approaches to solving problems of distortion, calibration, and object detection. We then dive into

our approaches to each of these three problems, and how they tie together into an end-to-end approach that could be deployed into retail stores. Finally, we use a larger scale dataset to obtain some quantitative metrics with which to evaluate the success of our approach, and we leave space for future work, such as integrating Extended Kalman Filters to enforce temporal consistency.

2. Problem Statement

Our objective is to accurately predict the 3-D position of a person based on their location in a security camera. If the person's true location (we use the location of their feet) can be given by $(x*, y*, Z*)$, and we estimate location (x', y', z') for them in 3-D space, then we are trying to minimize:

$$d = \sqrt{(x* - x')^2 + (y* = y')^2 + (z* - z')^2}$$

for each person in each image. Since we use people's feet, we can constrain $z' = 0$, enabling us to use a single view to predict position - this creates errors when people jump, but this is not typical behavior.

We use two coordinate systems in this paper. The primary system is standard, where x spans the width of the image, increasing to the right, and y spans the height, increasing downwards, and the origin $(0, 0)$ in the upper left hand corner of the image. We also use a unique coordinate system when undistorting images, where $(0, 0)$ is at the optical center of the image, and x and y increase right and downwards respectively. This coordinate system is necessary to model radial distortion parameters by expressing points in polar coordinates from the optical center.

3. Related Work

There is a significant amount of work that has been done in the space of retail analytics via camera, but this work has been done almost exclusively by startups which protect their methods as intellectual property. These include Prism Skylabs, Brickstream, and RetailNext. These are all dependent on custom hardware, or sensors to augment the surveillance feed.

On the technical front, we had to integrate work from various frontiers in computer vision. Solving this business problem required us to solve a number of technical problems. One was correction of barrel distortion - we leaned heavily on Sing Bing Kangs work in Semiautomatic Methods for Recovering Radial Distortion Parameters from A Single Image, in which he defined an algorithm by which a user to draws snakes on a distorted image, with each approximately corresponding to a projected straight line in space [8]. In his paper, he outlines how these snakes can be used consistently with a model of radial image distortion to solve for the radial distortion parameters and thus undistort the image. For a given snake, the algorithm fits it to the line of best fit, rotates this line to be horizontal, and estimates constant distortion parameters that fit all of these snakes/lines.

Another problem that was clear was recovering both intrinsic and extrinsic camera parameters from a single view. To do this, we used the affine calibration approximation taught in class, and covered in R. Hartley and A. Zissermans textbook, Multiple View Geometry in Computer Vision [7]. In particular, we used calibration from a checkerboard with the direct linear transformation algorithm, with tiled floors as our checkboard. We had also tried single view metrology with three sets of parallel lines, but this left us estimating extrinsics.

Finally, we had the problem of object detection, to find people in our image frame. Cutting edge research in object detection suggests that deep convolutional nets is the best way to do this. Scalable Object Detection using Deep Neural Networks by Erhan et al. at Google demonstrated that convolutional neural nets are very powerful for finding regions of interest, while also having an effective recognition path that categorizes the object of interest. The two steps take a while though, and are not necessarily fast enough to build real-time bounding boxes on video - Ren et al.s Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks takes this work a step further by folding the localization and recognition path into the same convolutional neural networks, training the weights with the localization and recognition cost functions alternately [10]. We use this work directly as a component of our solution.

Finally, after proving our concept on some footage via YouTube, we were able to find a much more expansive dataset with ground truths from A new Dataset for People Tracking and Reidentification via the Video Surveillance Online Repository [11]. This dataset was also precalibrated and undistorted for us, via the methodology outlined in Cooperative Object Tracking with Multiple PTZ Cameras, presented by Everts, Jones, and Sebe [3].

4. Technical Approach

4.1. Distortion Correction

While there are a number of approaches to fixing this issue, such as un-distorting the image with projections of area, or computing radial distortion coefficients, most methods depend on knowing the intrinsics of the camera predistortion. Sing Bing Kangs work, however, suggests a method to manually pick points on a line and accordingly fit distortion parameters, as referred to above [8]. In particular, it tries to fit all points that should be collinear (as denoted by a person) so that they are, while moving those

points as little as possible, and only moving all points radially by adjusting radius parameters.

Radial distortion, of which barrel distortion is a type, can be modeled by imposing a polar coordinate system on the image. From the center of the image, each pixel has an angle and distance from the optical center. Changes in this distance create radial distortion. The distortion at a point, which we call Δr, is the change in distance from the optical center from the undistorted distance. We model this distortion with the equation:

$$\Delta r = \sum_{i=1}^{\infty} C_{2i+1} r^{2i+1} \text{ where } r = \sqrt{x^2 + y^2}$$

Then, the approach is to find values of C such that all points we manually constrain to be collinear as collinear, while also minimizing the distance we move them. This is a common radial distortion correction algorithm, and we found that Photoshop actually provides a very effective implementation which can be used to adjust entire videos, and imposes the same distortion parameters on each frame. We went this route to manually undistort video, rather than implement the polar geometry and solver for the parameters from scratch.

4.2. Calibration

Being able to map pixel coordinates to world coordinates is a central component of understanding shoppers' 3D locations from images. Thus it is necessary to find a robust camera calibration for any given video feed. We considered two approaches to solving this problem for our unlabeled retail store data. For our 3DPeS data, calibration parameters were included with the dataset. The parameters were given in a third type of calibration formulation, which we also explain below. Because retail cameras only need to be calibrated once, it is practical to do these calibrations by hand in a real-world context; thus we did not invest time in automating the calibration process.

4.2.1 Single View Metrology

Given that we are constrained to one camera, in an environment with multiple sets of mutually orthogonal lines, it is natural to first try an approach based on single-view metrology to calibrate the camera. In our retail video, we labeled three sets of mutually orthogonal lines by selecting two points on each of 6 lines. For each pair of parallel lines, we found the corresponding vanishing point $v_i, i \in \{1, 2, 3\}$ by computing the intersection of the lines in image coordinates. We then considered the matrix ω, the projection of the absolute conic Ω_∞ into image coordinates. By assuming a camera with square pixels and 0-skew (a reasonable assumption for a retail security video that has been corrected for barrel distortion), we can constrain ω to:

$$\omega = \begin{bmatrix} \omega_1 & 0 & \omega_2 \\ 0 & \omega_1 & \omega_3 \\ \omega_2 & \omega_3 & \omega_4 \end{bmatrix} \quad (1)$$

This matrix has four unknowns, but it is only known up to scale. This means there are effectively three unknowns if we set one of the unknown variables to 1 and scale the rest accordingly. As a result, we can solve for the matrix ω by using our three vanishing points, and exploiting the fact that because they are mutual orthogonal, for each v_i, v_j with $i \neq j$, $v_i^\top \omega v_j = 0$. Thus we have three scalar equations in three unknowns:

$$v_1^\top \omega v_2 = 0 \quad (2)$$

$$v_1^\top \omega v_3 = 0 \quad (3)$$

$$v_2^\top \omega v_3 = 0 \quad (4)$$

It is known that $\omega = (KK^\top)^{-1}$, where K is the 3x3 matrix of camera intrinsics. So we can find K using the Cholesky decomposition of ω. We did this with our retail video and found the camera intrinsics. Unfortunately after this process the extrinsics $\big[R|T\big]$ are still unknown, so we could not recover the entire camera matrix $P = K\big[R|T\big]$. Setting the camera to be the origin in world coordinates is not helpful, because even though it resolves the $\big[R|T\big]$ parameters (they would simply be $\big[I|\mathbf{0}\big]$, we still need to know where the ground plane is in world coordinates to resolve the projective ambiguity of mapping a pixel to a world point. We tried estimating $\big[R|T\big]$ by hand through trial and error, but the results were very unreliable.

Figure 1. Sets of mutually orthogonal lines used for single view metrology calibration in the retail camera frame. The line intersections give us three vanishing points, which we use in the equations above to solve for K.

wher e

4.2.2 Affine Calibration

Our problems with a calibration based on single view metrology could be resolved by finding point correspondences and solving for the camera matrix directly. For our retail video, we labeled 15 points by hand in the scene. We place the origin at the bottom left corner of the bottom-leftmost tile that is fully visible, we let each tile be 1×1 in width and height in world coordinates, and we say that all tiles lie on the ground plane $z = 0$. We model the camera matrix P as affine, which is a desirable approximation even though the true camera matrix is projective, because the lines in the scene are nearly parallel and solving for fewer unknowns is preferred with only 15 point correspondences. That is, we let:

$$P = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

Then, we use our $n = 15$ points to solve the following over-constrained system of $2n$ equations:

$$A\mathbf{x} = \mathbf{b} \tag{6}$$

Where the world coordinates of point i are (x_i, y_i, z_i), the image coordinates are (u_i, v_i), and:

$$A = \begin{bmatrix} x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 \\ x_2 & y_2 & z_2 & 1 & 0 & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ x_n & y_n & z_n & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 \\ 0 & 0 & 0 & 0 & x_2 & y_2 & z_2 & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & x_1 & y_n & z_n & 1 \end{bmatrix} \tag{7}$$

$$\mathbf{b} = \begin{bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ . . \\ u_n \\ v_n \end{bmatrix} \tag{8}$$

And $\mathbf{x}$ is a column vector of the eight unknowns in P, arranged in order with unknowns from the first row of P before unknowns from the second. We solve the system of equations in the standard way, by rearranging so that the right hand side is $\mathbf{0}$ and the matrix A has additional columns (and $\mathbf{x}$ additional rows) to maintain the constraints imposed in the original equation by the values in $\mathbf{b}$; then taking the SVD of this augmented left-hand matrix, and using the last column of the third output from SVD as the parameters of P.

4.2.3 PTZ Calibration for Ground Plane Object Detection

Affine calibration worked well for our retail video data. But we used a different approach when working with the 3DPeS dataset, because that dataset already included parameters for a different type of calibration. Due to the difficult position of the cameras in the dataset, the publisher used a simpler type of calibration [3] designed specifically for Pan, Tilt, and Zoom (PTZ) cameras, which are commonly used in surveillance. The methodology is fully described in the source paper; we briefly summarize it here for convenience.

The calibration assumes that objects are only detected along a ground plane of $Z = 0$. Let U, V, H be the displacement of the camera coordinate system relative to the world; $\Delta i = i - i_0, \Delta j = j - j_0$ are the pixel positions relative to the image's optical center (i_0, j_0); α_x^f and α_y^f are the horizontal and vertical scales between the image and image plane; t is the tilt angle of the camera; and $p' = p + p_0$ is the pan angle after the camera is aligned with the world coordinate system. An object's world coordinates X, Y are then given as:

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \frac{H}{\alpha_y^f \Delta i \sin t + \cos t} R \begin{bmatrix} \alpha_x^f \Delta j \\ \alpha_y^f \Delta i \\ -1 \end{bmatrix} + \begin{bmatrix} U \\ V \end{bmatrix} \tag{9}$$

where

$$R = \begin{bmatrix} \cos p' & \sin p' \cos t & \sin p' \sin t \\ \sin p' & -\cos p' \cos t & -\cos p' \sin t \end{bmatrix} \tag{10}$$

4.3. Person Detection

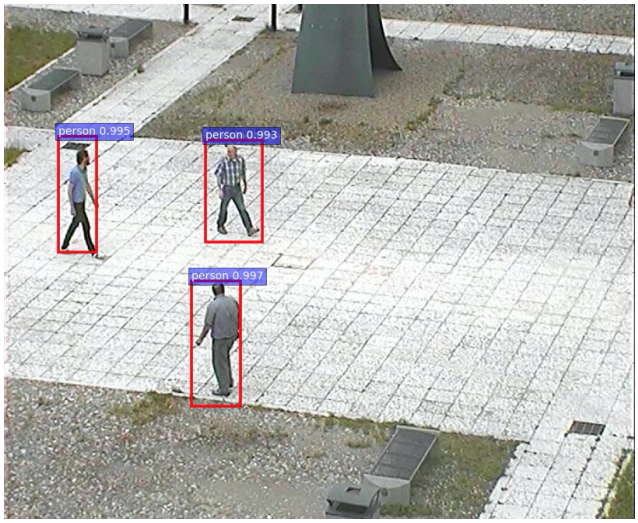


Figure 2. Our ConvNet can detect multiple pedestrians with high confidence, especially in clear environments such as this one. This image is from Camera 3 in the 3DPeS dataset.

Person detection in 2D images is a well-studied problem [12] [1] [6] with several existing solutions, offering various tradeoffs between speed, accuracy and simplicity. There are essentially two parts to the problem: generating regions of interest (RoIs) where a person might be, and classifying those regions to determine if the region does indeed contain a person.

Recently, approaches that utilize deep ConvNets have been shown to perform exceptionally well at object detection, and person detection specifically. For this part of our problem, we use a deep Convent object detection architecture proposed by Girshick et. al. known as Faster R-CNN [10]. Faster R-CNN is an improvement on Fast R-CNN [4], which is itself an improvement of the original R-CNN architecture [5]. Faster R-CNN works as a single, unified ConvNet that uses shared convolutional layers to output feature maps that then get sent to a Region Proposal Network (RPN) and a classifer head. The network is trained end-to-end with back propagation and stochastic gradient descent, with a multi-task loss function. The full details can be found in [10].

We use a pretrained version of Faster R-CNN that we modified to only output person detections (the original version outputs detections of 20 types of objects).

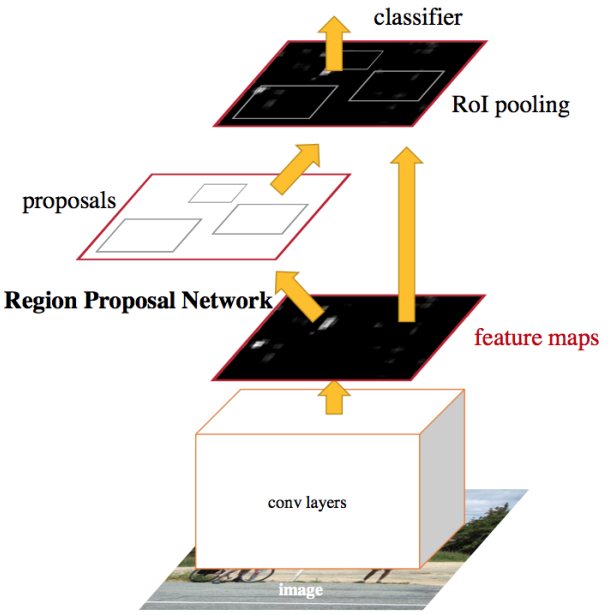


Figure 3. The Faster R-CNN architecture. (Image from [**?**]ref3))

4.4. Putting it Together

Given a neural network that can detect 2D bounding boxes around people and a calibrated camera, how do we put the system together? Our approach takes as input a stream of video data. For each frame in the stream, we run our person detector over the image and get as output a set of bounding boxes. In the general case, the mapping between the pixels defining the bounding box and the world coordinate system is ambiguous, thanks to the ambiguity of the 3D to 2D transformation of the camera. But because we know these bounding boxes are people, we can make an assumption that each person's feet rest on the ground plane ($Z = 0$). This is a reasonable assumption in nearly all retail environments; the only test videos we encountered in which this is not the case is when the camera watches over an escalator or can see multiple floors at once.

Once we have bounding boxes for each person, we take the bottom center pixel of each box (call its image coordinates c_x, c_y) and find the 3D coordinates associated with that pixel, assuming it lies on the $Z = 0$ plane. In the affine calibration case, this means finding the intersection of the ray from the camera along which any point in 3D would project to (c_x, c_y), and the $Z = 0$ plane. By construction this intersection must resolve to a unique point.

In the PTZ calibration case, the ground plane assumption is built into the calibration model and so there is nothing else that must be done besides converting c_x, c_y to offsets from the optical center and plugging the results into equation (9).

In both cases, once we have obtained the world coordinates of each person, we plot 3D voxels representing each person in 3D graphing environment modeled after the room, to visualize the positions of the people in 3D.

5. Experimental Setup and Results

We performed experiments on two data sources for this project. The first was from a video clip of a retail security system demo on YouTube [2]. This clip was useful for us to understand real world surveillance video conditions. For example, before looking for real surveillance video online we had not considered the fact that we might encounter barrel distortion: upon encountering that problem we realized a practical implementation would need to correct for this (which we now do). It was also helpful as a qualitative assessment tool of our system's performance. Unfortunately YouTube videos don't have ground truth labels, so we could not evaluate our results quantitatively with this data source.

The lack of truth labels led us to search for other data sources. We found that the 3DPeS Video Surveillance dataset was quite useful in this regard. This dataset contains outside surveillance footage, where there are often fewer occlusions and people are farther away than the in-store environment, so it is not exactly representative of a retail system. Nonetheless the dataset offered numerous advantages, including ground-truth labels and pre-computed calibration parameters. It gave us the chance to address the same fundamental task, predicting where people are in 3D given a

single 2D camera, in a cleaner environment.

5.1. Environment

We performed all tests on late-2013 Macbook Pro with 16GB RAM and a 2.3GHz processor. Due to lack of hardware, we ran all code on the CPU, even though the ConvNet runs much faster on the GPU. This resulted in an average execution time of 3.19s per frame, a roughly 15X slowdown in prediction speed compared to the results reported by [10] on better hardware. The time spent outside of our ConvNet's forward pass was negligible. From these numbers it is clear that a real world deployment of our work should have a dedicated GPU.

5.2. Results and Error Analysis: Retail Clip Experiments

Figure 4. Example bounding box predictions for the retail video data. These were generated without doing image distortion correction, although in our final implementation we were sure to correct distortion first if it was present.

The goal of our experiments on the retail clip data was to verify qualitatively that our approach was sound, and to produce for each frame a 3D visualization of the scene geometry with people accurately tracked throughout. In our affine calibration step, we hand-labeled 15 point correspondences, shown here in figure 5. The root-mean-square error (RMSE) of the calibration matrix we found on the data used to create it was 32.7084 pixels, less than the width of one tile almost everywhere in the frame. Ad-hoc measurements of the final 3D voxels outputs showed they were generally within two thirds of a tile to the true position of each person when the bounding box was correct, or roughly 20cm. We did not analyze this rigorously, as we performed most of our quantitative analysis on the second dataset.

A common failure mode of our solution on the retail clip data was occlusions. Occlusions cause problems in two ways. The first is that they sometimes prevent our ConvNet from finding a person in the frame. Even if the ConvNet does find a bounding box, however, occlusions can still cause problems if the feet of the person are not visible in the image. This is because our pipeline assumes that the bottom of the bounding box is where a person's feet are, and thus where the ground plane is. When that assumption is violated (e.g. because the bounding box ends at the person's waste), then the output is noticeably inaccurate.

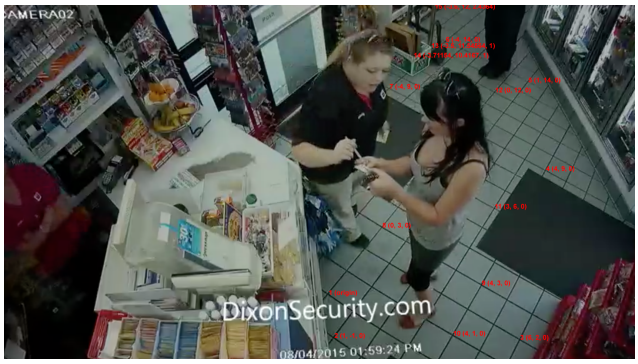
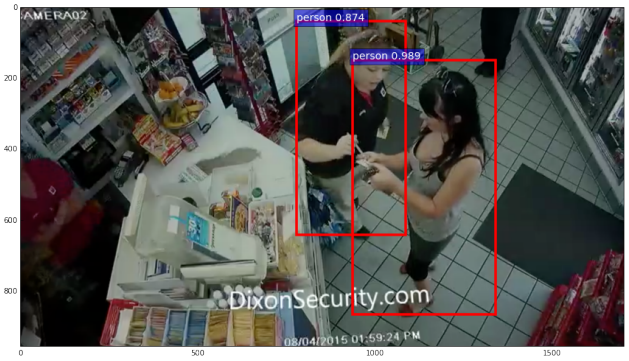


Figure 5. The point correspondences we labeled for the retail camera video clip, vantage point 2. It is important that some of the points are off the ground plane, or the calibration would be degenerate.

Another typical failure occurred when people were in rapid motion. In the video clip, there is a point at which the two women sprint out of the store. For most of these frames, the system loses track of them because no bounding boxes are predicted. We hypothesize two reasons for this failure. One is that the rapid and blurry stills of a human sprinting do not look very much like a typical person, and these types of images are likely underrepresented in the dataset on which our ConvNet was trained. The second is that due the underlying architecture of the ConvNet, it has a receptive field size of 228 pixels. This is suitable for most purposes but when the people in this video clip are sprinting with arms extended on both sides, their width in the image easily exceeds 400 pixels. This makes it nearly impossible for the ConvNet to have a chance at detecting the entire bounding box.

5.3. Results and Error Analysis: 3DPeS Dataset Experiments

We also evaluate our pipeline on the 3D People Surveillance Dataset provided by [5]. In general our people detection ConvNet works much more reliably on this dataset because of the reduced occlusions, better lighting and higher definition of the images. In our run of the pipeline on a live stream of 17 frames from the same camera, we detect 51 of 53 total person bounding boxes when the person is more than halfway in the scene (i.e. not majority cut off by an edge of the image). Across all frames we tested, the root-mean-square error of our position predictions in world coordinates was 554 millimeters. This is about 2x higher



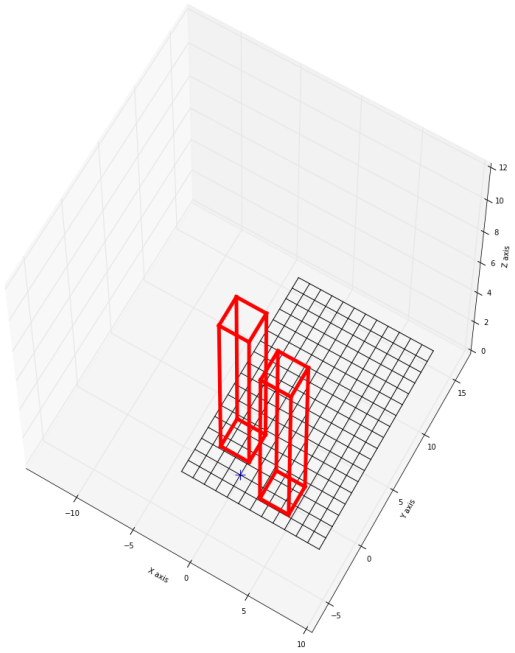


Figure 6. Bounding boxes found for a sample frame and the corresponding 3D scene model that we generated. In the 3D model, the origin is marked by the blue plus sign below the left voxel. It corresponds to the tile in the frame found right below the "i" in "DixonSecurity.com". We can see here that the model is rather accurate considering the low number of calibration points, the original fisheye distortion, and partial occlusions in the scene.

than our ad-hoc estimate of our performance on the retail dataset, due mostly to the vastly greater field-of-view of the camera we used in this dataset. (This is an outdoor camera which overlooks more than 200 square meters of space, much more than can be seen by the indoor camera. So being off the same number of pixels will translate to a much larger increase in RMSE.)

We can glean several interesting insights from the prediction error graph. For example, we see that in general

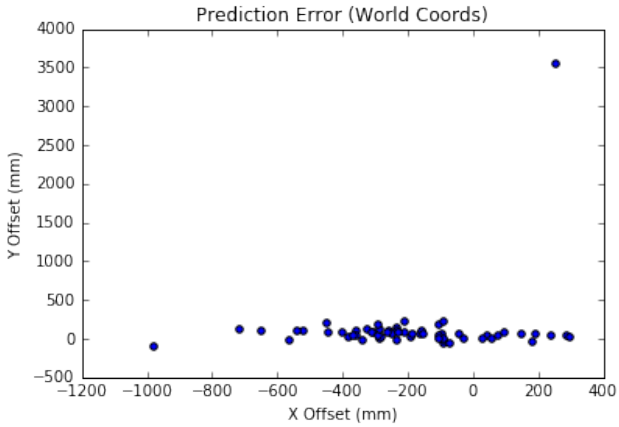


Figure 7. Our prediction errors in millimeters in the world frame for each person in each image we evaluated, shown collectively. Each point is the difference between the predicted x, y of a person in world coordinates and the true x, y of the person.

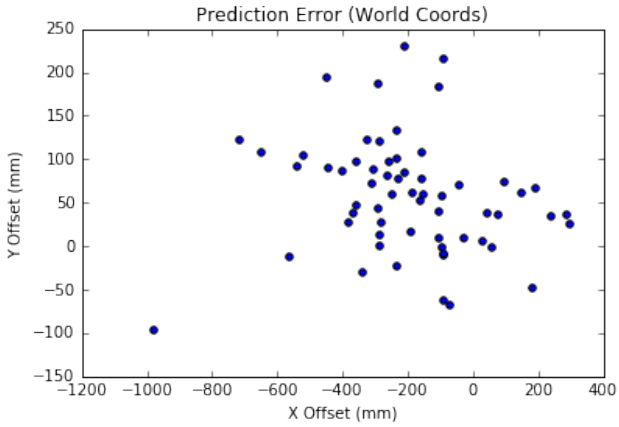


Figure 8. The same graph as before but with the outlier (Y offset > 3000) removed. Note the different scales of the X and Y axes.

there is more error along the X axis than the Y axis, but most of the Y axis error that does occur is in the same direction: consistently slightly positive. This is because we use the bottom of the bounding box as the intersection point of the person with the ground, when in reality the ground truth label for the person's position in 3D considers the center of the person overall. (Imagine a circle on the ground around the person's feet. The centerpoint of this circle is the ground truth x, y label. It will consistently be slightly offset from a point at the edge of one foot, which is what we get with the bounding box method.)

The X axis error is also because of our bounding-box-to-intersection-point methodology. As people walk they swing their arms and stride their legs. The bounding box produced by our ConvNet will generally capture all of these extremities, so any time they are not displaced from the person's

center symmetrically, the bottom center of the bounding box will not be an accurate representation of where the feet intersect. Finally, the bounding boxes are in general imperfect, and random noise is surely a factor as well.

6. Conclusions and Future Work

We've successfully developed and outlined an end-to-end approach to turning raw security footage into a 3D model of customer movement throughout a retail store. This involves a one-time calibration of distortion and camera parameters, and then the usage of Faster-RCNN to find people in the frame. The aforementioned parameters are used to related their location on frame to their real location inside the store. While scaling difficulties arise in the once-per-deployment cost of manually determining the camera's distortion, intrinsic, and extrinsic parameters, this method seems to be accurate enough to effectively provide data to retail environments.

With typical error in the range of 20cm or so in real space in indoor settings, this could very plausibly be used to track the location of shoppers in a retail space - information such as aisle choice, for instance, is easily determined at this level of granularity. Back projecting the person's location into 3-D space is very important for these businesses, and it's exciting that a simple and practical assumption about position (that feet are on the ground) is so effective.

There is, however, a major obstacle to usage of this approach in practice. This is occlusion of the feet - it's not uncommon for shelves or other objects to block the feet of subjects, making it impossible for our existing algorithm to guess their position. We had to carefully select datasets because of this limitation, but real retail stores will not be able to do this, instead having to work with whatever their camera sees. One promising way to handle this would be to use temporal consistency (i.e. relate similar bounding boxes across timeframes) to estimate foot position now based on foot position in previous frames. This could be done with Extended Kalman Filters, which allow us to integrate a physical model of the world alongside noisy measurement data (the person detector) to produce an output that is overall more robust. We can also use a Faster-RCNN architecture ConvNet trained specifically to look for feet, and when feet are not detected in a bounding box (due to occlusions) we can instead use the position of the face and extrapolate downward based on assumptions about human proportions. The problem of consistent offsets in one direction caused by the edge-of-feet point from the bounding box v.s. the between-the-feet ground truth point can be resolved with a simple addition of a mean error vector x_Δ, y_Δ that can be learned from training data. Overall, we think this is a compelling first step towards real-time 3D person detection in retail and that the remaining obstacles are surmountable.

References

[1] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. *Computer Vision ECCV*, 2006.

[2] T. Dixon. Fight caught on cctv security camera. `https://www.youtube.com/watch?v=Kla8W8IIAtk`.

[3] I. Everts, G. Jones, and N. Sebe. Cooperative object tracking with multiple ptz cameras. *Image Analysis and Processing*, 2007.

[4] R. Girshick. Fast r-cnn. *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, 2013.

[6] I. Haritaoglu, D. Harwood, and L. S. Davis. W4s: A real-time system for detecting and tracking people in 2 1/2d. *Computer Vision — ECCV*, 1998.

[7] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[8] S. B. Kang. Semiautomatic methods for recovering radial distortion parameters from a single image. *Technical Report CRL*, 1997.

[9] F. Manzella and I. T. Teije. The truth about in-store analytics: Examining wi-fi, bluetooth, and video in retail. 2014.

[10] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv*, 2015.

[11] V. S. O. Repository. 3dpes: A new dataset for people tracking and reidentification. `http://imagelab.ing.unimore.it/visor/3dpes.asp`.

[12] Z. Zivkovic and B. Krose. Part based people detection using 2d range data and images. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.