Table of Contents

Table of Figures and Tables

Introduction

Background and Motivation

Physical exercise is a key contributor to good health. Despite this, two out of three Americans do not meet the recommended minimum amount of weekly exercise. According to the International Health and Racquet Sports Association, difficulty in receiving immediate feedback on one's workout is a principal driver of gym attrition (Pernek, 2013). One way to address this problem of limited feedback in the gym is to create a workout log that allows users to observe trends in their exercise history. Unfortunately, current products in the market are all either burdensome or are too limited in scope. See Appendix A for a case study of available products.

Outside the market, there have been recent academic efforts to develop exercise classification and repetition counting capabilities. Most approaches focus on accelerometer data (Li, 2013) (Sundholm, 2014) (Velloso, 2013) and some have found machine learning techniques to be a practical option for exercise classification (Novatchkov, 2012). As our ENPH 459 project, *IoT Gym*, we developed a system that uses machine vision to automate the classification of exercises, counting of repetitions and detection of weights. This system has the potential to be more versatile than mounted accelerometers. The *IoT Gym* project identified a camera system as the most viable option, after considering other sensor-based solutions. Using a single Microsoft Kinect V2 camera, the project succeeded in detecting a single user performing any of eight different free or body weight exercises, as well as counting the number of reps performed and determining the weight used. It does this by generating a skeleton of the user consisting of 25 joints. A machine learning algorithm then processes this skeleton to determine the exercise and count repetitions. See Appendix C for an IEEE journal style paper summarizing the technical details of our previous work.

This project, *Skeletal Capture*, aimed to improve the quality of generated skeletons used in exercise classification and repetition counting, as well as create a system that could utilize multiple cameras to cover a wider field of view. It was also important to build a system that used modern cameras, such as Intel RealSense cameras, as Microsoft has discontinued the Kinect camera range and is no longer selling them (Lin-Poole, 2017).

Project Objectives

In our initial project proposal, we outlined the following objectives:

Original Core Goals

1. Demonstrate improved accuracy of skeleton data generated with multiple Intel RealSense cameras over the Kinect. This will include direct side-by-side visual comparisons. This is particularly important for body positions proven to be troublesome with the Kinect, including push-ups and sit-ups.
2. Perform a cost to areal coverage analysis for the original Kinect and the new skeletal capture system. That is, determine Total Cost ($) / Total Area Covered ($m^2$). Total cost is the cost of acquiring all components new, and total area covered is the total contiguous area a human test subject can travel while the system is still capable of capturing the subject's skeleton and motions.
3. Estimate the total cost for the system to be installed in a commercial gym. This includes installation, operation and maintenance costs.

Original Secondary/Stretch Goals

These 4 goals primarily focus on features that are requirements for implementing a skeletal capture system in a commercial setting like a gym. Most of these goals build upon the primary goals.

1. Using data from multiple cameras, detect skeletons around obstructions. This is a vital part of implementing a commercial system, as there will surely be obstructions.
2. Implement a system capable of visualizing captured skeletal data in real time.
3. Track multiple people over cameras with different fields of view. In a setting such as a gym, many people will be operating near one another, and the system must be able to differentiate between individuals, as well as track all skeletons in motion.
4. Perform exercise detection on improved skeleton data.

New Goals

Due to technical issues that arose while we were pursuing the first primary goal, the goals of this project pivoted to:

1. Create method to automatically calibrate a multi-camera setup. The relationship between all camera coordinate systems need to be determined to derive a set of mappings. These mappings can then transform a point cloud from each camera to a global coordinate system. A method to easily do this will allow future work to be much more efficient.
2. As a stretch goal to create a more compelling demo, generate a user's skeleton from the point cloud.

Scope and Limitations

The original scope of this project was to improve the accuracy of skeleton data collected by the Kinect using multiple RealSense cameras and to demonstrate this improved accuracy. Throughout the course of this project, however, this scope decreased to developing a method to calibrate a multi-camera setup, and visualizing data from the calibrated system in the form of a point cloud.

Report Organization

The remainder of the report is structured under several sections. First, we begin with a technical discussion section, under the heading Discussion. This section begins with a brief overview of some of our initial results, which led to a change in focus for our project. The rest of the Discussion section is written from low to high levels of abstraction. We begin with the mathematical foundation of our work, and then build up to a system level description. The Discussion section ends with a summary of tests we developed for our designs and their results.

Following the Discussion are the Conclusions, Recommendations and Deliverables sections. These sections discuss the conclusions we can draw from the tests we conducted and potential next steps in the development of a skeletal capture system for automated workout tracking.

Discussion

Our first approach to generating skeleton data was to use multiple RealSense cameras and involved generating a skeleton at each camera. A combination of skeletal data from multiple cameras would then allow us to generate a better one at a central hub. We identified Nuitrack as the best third-party library to generate a skeleton from a single camera due to the low processing power required, which meant that we could use cheaper hardware, while still meeting the frame rate requirement for detecting quick motions in exercises. After looking at the example viewer provided with Nuitrack, however, we realized that the skeleton data was poor in two important cases. This includes both when the user was beyond a distance of 4 meters and when the user is in different positions such as performing a push-up or sit-up. See Figure 1 for examples.

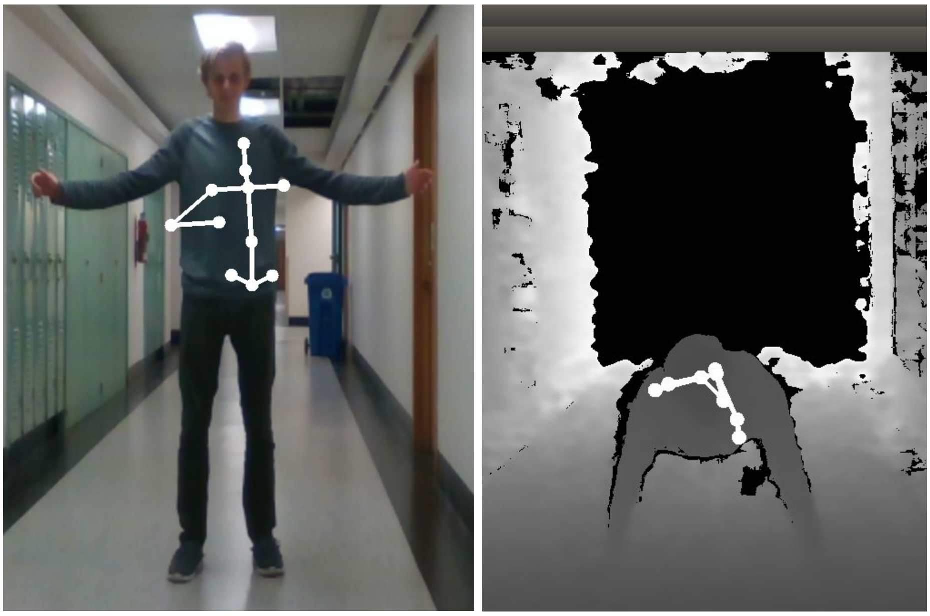

Figure 1. Legless skeleton just beyond 4 meters (left), and inaccurate skeleton when performing a push-up (right)

The poor quality of this skeletal data generated by a single camera meant that our original plan of generating a skeleton at each camera and then combining them centrally would not be viable. Instead, we hypothesized that if we could combine data from multiple cameras into a single point cloud, the extra information would allow for improved skeleton generation. This result is what lead us to pivot our objectives, instead focusing on creating a calibration system and visualizing point clouds, which is what the rest of the discussion focuses on.

Theory

A point cloud is a set of data points in a three-dimensional space. They are an effective means of imaging objects by using many points to represent an external surface. The RealSense D415 cameras used in this project consist of a stereoscopic depth camera and an RGB camera in a single package. Each one produced a two-dimensional image. The RGB sensor produces a conventional image that stores a colour mapping for each pixel, while the other sensor stores a depth value in the camera's coordinate system.

There are two major calibration components to consider in the process of going from depth and colour images to a full three-dimensional point cloud. The first is intrinsic adjustment, which takes into account manufacturing differences between cameras, and is used to convert the two images into a point cloud in a coordinate system specific to the camera. The second is extrinsic alignment, which adjusts for the relative position of multiple cameras, and is used to map each camera’s point cloud to a global coordinate system. The theory behind these two processes is outlined below.

Intrinsic Matrix

The first step in creating a depth image is to process information from the stereoscopic sensor. This sensor uses two parallel view-ports and a projected infrared dot matrix, and calculates depth by estimating disparities between the dot matrices in the two adjacent images, similar to how human binocular vision works. Once the sensor determines the depth, it is vital to consider corrections for intrinsic factors of the camera. Intrinsic factors include focal length, pixel size and the principal point of each camera (Heartly, 2003), all of which are slightly different due to manufacturing tolerances. All these factors can lead to distortion of a point cloud causing a systematic deviation of that point cloud from the real-world object it is trying to represent.

We can calculate intrinsic camera parameters by imaging known objects in various positions. In this project, we used an Intel RealSense tool to calculate these parameters using a checkerboard like target (Intel, 2018). Once we calculate these parameters, we can use them to correct the raw depth data generated by the stereoscopic sensor. The intrinsic matrix describes the relationship between the uncorrected coordinates (u, v, w) and the corrected coordinates, (x, y, z). The matrix is given below, where f_x, f_y are focal point positions and p_x, p_y are pixel dimensions.

$$\begin{bmatrix} -f_x & 0 & p_x \\ 0 & -f_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

We omit a formal derivation of the above expression, as it is not vital to the understanding of this project, but an interested reader can find a derivation in Simek (Simek, 2013).

Extrinsic Matrix

One key problem in reconstructing a single point cloud from individual point clouds generated by cameras with different perspectives, is that each point cloud has its own distinct cartesian coordinate system.

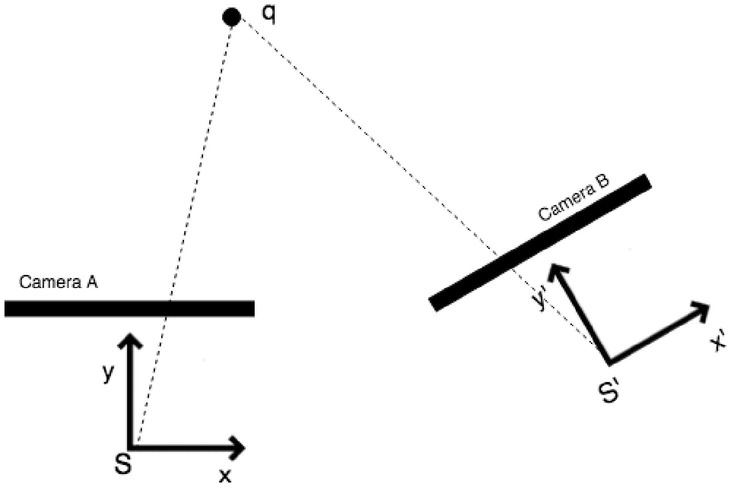

Figure 2. A two-dimensional depiction of the two-camera problem. Both cameras can see the point q, but it corresponds to different points in the S and S' coordinate systems. S' can be mapped to S with only a rotation and a translation, otherwise known as a rigid transform.

Let us consider the case of two-cameras. Let S be the coordinate system of camera A, and S' be the coordinate system of camera B. Each camera images an arbitrary point q within the field of view of both cameras and record its depth. In coordinate system S, q corresponds to the point m, while in S' q corresponds to the point m'

$$m = [x \quad y \quad z \quad 1]^T$$
$$m' = [x' \quad y' \quad z' \quad 1]^T$$

In the definition of these vectors, we add 1's for convenience during transformation. Let the subscript 0 denote the same vector without the added 1.

$$m_0 = [x \quad y \quad z]^T$$

If intrinsic effects are perfectly accounted for during the construction of the image, we may assume that a single rigid transformation would allow us to perfectly map all possible points m' in S' to the corresponding point m in S. A rigid transformation T has the form:

$$(1) \quad T = [R|t]$$

Where R is a three-dimensional rotation matrix, and t is a translation vector. This type of transformation is the only type of transformation one can apply to a physical rigid body.

We seek such a transform T, known as the extrinsic matrix, that has the property:

$$(\ 2\ )\quad m = Tm'$$

An exact solution for T satisfies (2) under the constraints of (1). While finding exact solutions are possible, for the case where intrinsic corrections are perfect, we will omit it in this report in favour of a discussion on finding approximate solutions using methods that are robust against noise and inaccuracies in the intrinsic calibration and the raw stereoscopic data.

In practice, there are large inaccuracies in the imaging systems. To combat this, we gather a locus of points imaged with both cameras:

$$\mu_i' = [\,x_i' \quad y_i' \quad z_i' \quad 1\,]^T$$
$$\mu_i = [\,x_i \quad y_i \quad z_i \quad 1\,]^T$$

Giving us the corresponding equations:

$$(\ 1^*\ )\quad T = [R|t]$$
$$(\ 2_i^*\ )\quad \mu_i = T\mu_i'$$

Note that (2*) is now a family of equations. Instead of exactly solving (2*) we will have to compromise. We wish to find the best T, which satisfies the constraint (1) and minimizes the error:

$$e = \textstyle\sum_i |\mu_i - T\mu_i'|^2$$

We accomplish this in several steps. First, let μ and μ' be the matrices with μ_i^T and $(\mu_i')^T$ as rows. We begin by applying a multivariate regression model:

$$M = (((\mu')^T\,\mu')^{-1}((\mu')^T\,\mu))^T$$

Note that M is the operator which minimizes e, but does not in general, satisfy (1). Then performing a singular-value decomposition on the non-translational part of M:

$$M^{(1)} = \left[R^{(1)}|t^{(1)}\right]$$
$$R^{(1)} = S\Sigma V^T$$

Removing the central matrix from the decomposition gives a matrix that will be in the form of (1)

$$R^{(2)} = SV^T = R$$

This matrix $R^{(2)}$ will be the optimal rotation matrix sub-matrix for the rigid transformation. To determine the optimal translation sub-matrix:

$$t = \mu_0 - R^{(2)}\mu_0'$$

Following this procedure gives us the best rigid transformation. We can attempt to get better fits if we disregard the restriction (1). Without this restriction, the transformation is an affine transformation, which allows for stretching along any of the coordinate axes.

$$T_{\text{affine}} = M^{(1)}$$

This affine transformation allows for arbitrary linear transformations in any of the three-coordinate variable, but not stretching in higher orders of coordinate variables. If we change the form of μ, we can increase the degree of freedom in the mapping. There is no need to change any of the other formalism. For a non-mixed second order multivariate regression we use the form:

$$\mu_i = [\, x_i \quad y_i \quad z_i \quad x_i^2 \quad y_i^2 \quad z_i^2 \quad 1\,]^T$$

For a mixed second order multivariate regression we use the form:

$$\mu_i = [\, x_i \quad y_i \quad z_i \quad x_i^2 \quad y_i^2 \quad z_i^2 \quad x_i y_i \quad y_i z_i \quad z_i x_i \quad 1\,]^T$$

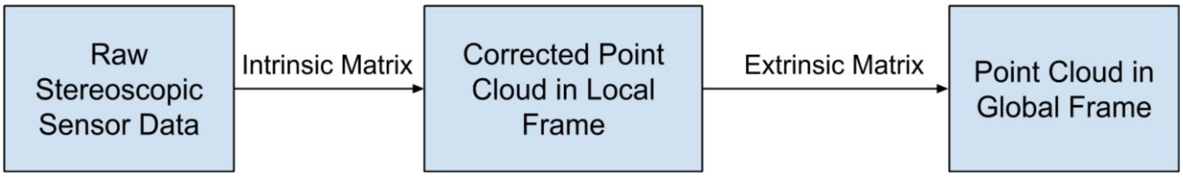
Raw Stereoscopic Sensor Data
Intrinsic Matrix
Corrected Point Cloud in Local Frame
Extrinsic Matrix
Point Cloud in Global Frame

Figure 3. Diagram depicting the processing flow of depth data from raw stereoscopic sensor data to the final point cloud. After the stereoscopic sensor generates the depth data, the intrinsic matrix corrects for focal length and pixel size. The extrinsic matrix then maps the local points in the point cloud to the global frame.

Design

Our test area was set up as shown in Figure 4. Four cameras cover a 4x3.5-meter area, with one at each corner. We perform the calibration data collection process as follows:

1. All four cameras start recording, saving data with intrinsic adjustments included

2. A user carries around a brightly coloured ball that is visible to most of the cameras for two minutes
3. All cameras stop recording
4. We run the calibration script, which finds the ball's center in each frame then performs pairwise extrinsic calibration

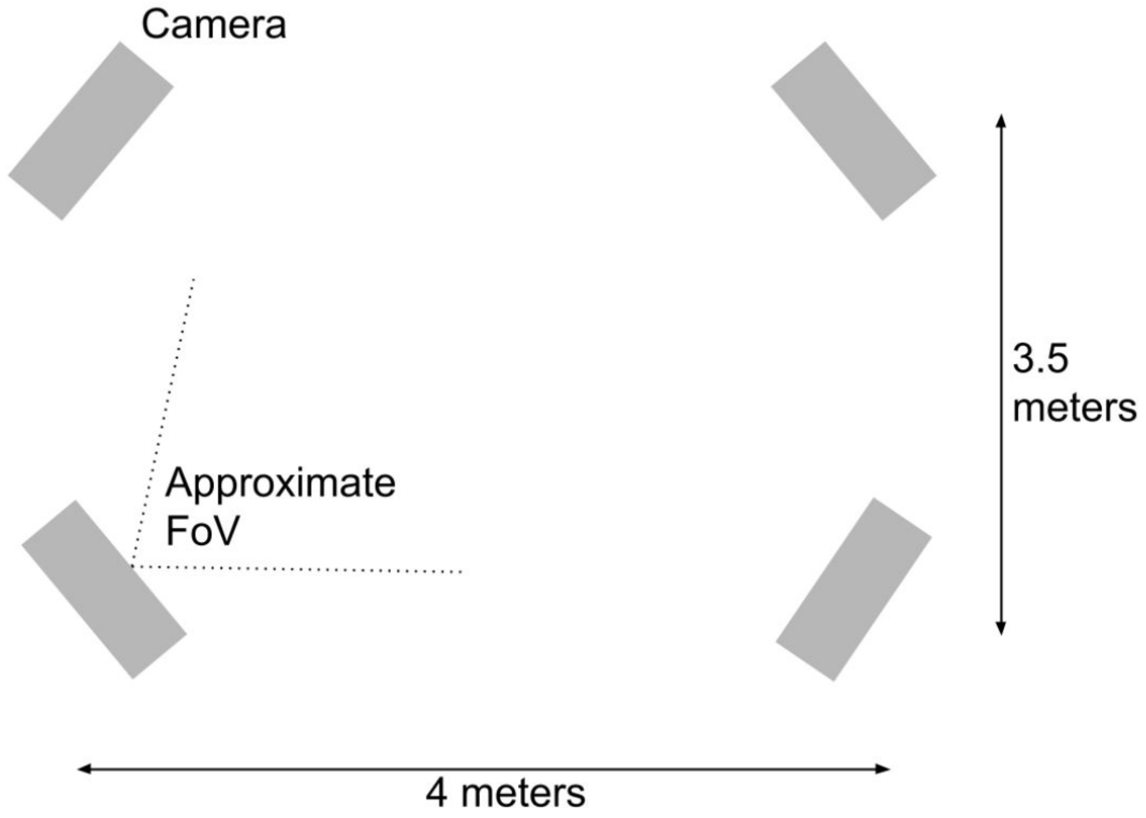

Figure 4. The arrangement of the four cameras in the room, with the approximate field of view (FoV) of one of the cameras. Exact FoV for the D15 Camera is horizontal: 69.4^o , vertical: 42.5^o, diagonal: 77^o (Intel, 2018).

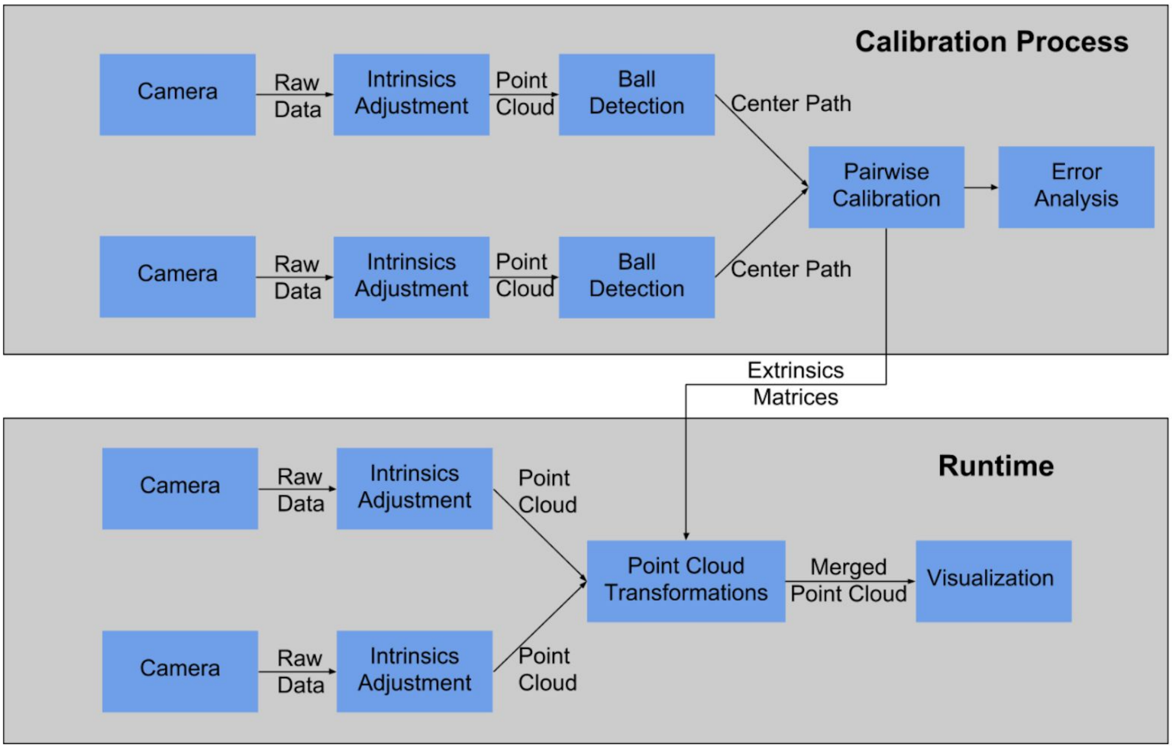

Figure 5. Data flow for calibrating two cameras against each other and then using the calibration during runtime. Intrinsic adjustment to the raw data generates a point cloud. Then a vision algorithm detects the ball in each frame. Then we perform a pairwise calibration on the path of the ball's center to generate an extrinsic matrix. This extrinsics matrix is used during runtime, and the full point cloud is visualized.

Intrinsic Calibration

To calculate intrinsic parameters of each camera we utilize the Intel RealSense Dynamic Calibration Tool (Intel, 2018). We followed the calibration procedure as outlined below:

1. Initialize the computer vision software capable of detecting a checkerboard pattern.
2. Move checkerboard pattern to various points in the field of view of the camera.
3. Perform image analysis of captured checkerboard patterns to determine intrinsic parameters of camera.
4. Upload intrinsic parameters to firmware of camera.

Despite the calibration, clear intrinsic errors prevailed in the collected point cloud data. These errors propagate through during the calibration and are one of the fundamental sources of error.

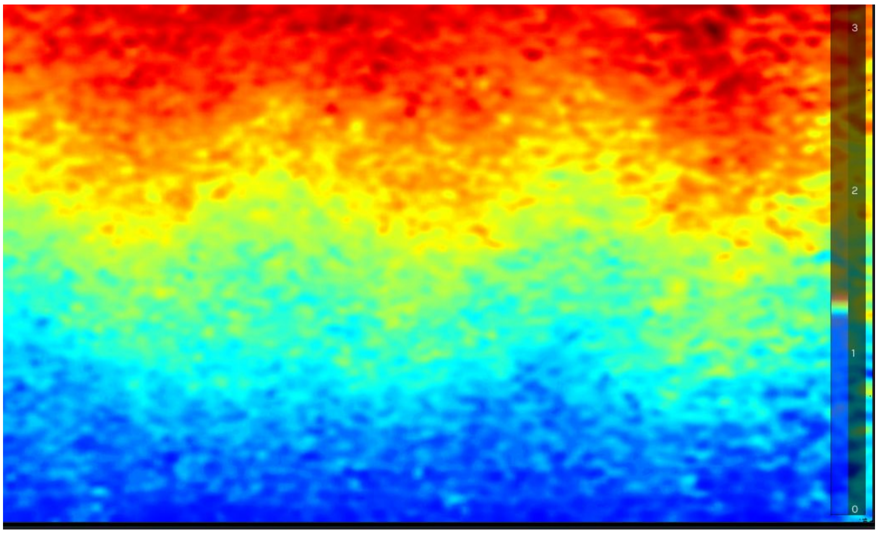

Figure 6. Calibrated depth camera imaging a flat wall, while placed at a 15-degree angle vertically. Waves in the depth information transverse to the depth gradient are visible. Despite repeated attempts at re-calibration, this pattern persists.

Extrinsic Calibration

To perform the extrinsic calibration, we needed to obtain many points that are known to be the same point in a global coordinate system but are different in each camera's coordinate system. To do this we followed the approach in Su (2018), which uses a large (approximately 30 cm diameter) brightly coloured ball that can be moved around and seen by all cameras. We can identify the ball in the RGB image using its colour and can determine the cartesian coordinates of the points on it's surface by mapping the RGB image to the corresponding depth image. We can then fit a sphere with the ball's radius to these points and can identify the coordinates of the sphere center. We perform these steps at all the cameras for each frame in the video, and then have a sphere center path in each camera's frame, which correspond to the same real-world points. We then use these paths to determine the transformations from one camera's coordinate system to another.

Ball Detection

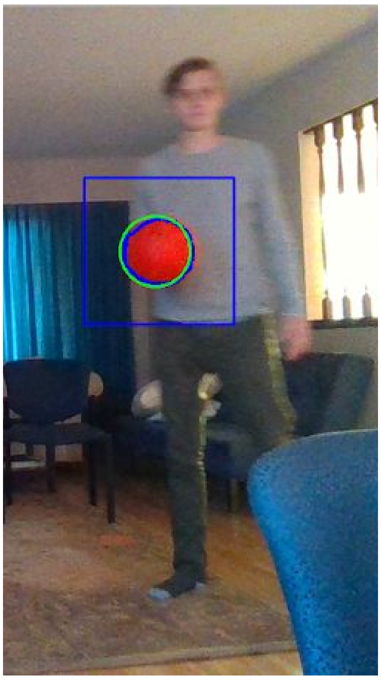

Figure 7. Contour around ball (blue circle), circle search box (blue rectangle) and identified circle (green). Note that the blurriness is due to the RealSense camera's slow shutter.

As shown in Figure 7, the orange ball stands out significantly in most scenes, making it easy to identify by looking for the largest clustering of pixels with a colour close to the ball's colour. To do this for a single frame, we first convert the RGB image to hue, saturation, value (HSV) space. The ball has a specific hue which is independent of the lighting conditions, and a high saturation, so we create a binary mask that selects only pixels that fit these criterion. Specifically, the hue must be within 8 of 178 and the saturation must be above 180 when using OpenCV's RGB to HSV transformation.

We then find the largest contour in this binary mask, which will be the ball unless there are any other large, similarly coloured objects in the room. We then search for circles in a rectangle that is 40% wider and taller than then contour. In Figure 7 the blue rectangle is the searching region. We then employ the circle Hough Transform inside the rectangle to find the circle. Finally, we perform a sanity check to confirm that this contour is the ball by taking the radius of the circle in pixels and, knowing the approximate angular coverage of a pixel and the average depth of points within the contour, can estimate the radius of the ball. If this estimate is within 15% of the known radius, then this contour is likely the ball, and we can fit a sphere and find the center coordinates for this frame. We repeat this process for all frames, outputting a list of center points and timestamps for use in the following steps.

Pairwise Mapping

With sphere center paths and timestamps for each camera, we can determine the transformations between any two cameras' coordinate systems using any of the methods described in the theory section. First, however, we need to interpolate the trajectories in time so that we have sets of points in each frame at the same times. Also, because the ball may not always be visible in each camera's field of view, we must determine the valid time ranges for each camera, and only compare points when the ball is visible to both cameras.

We specify a valid time range as a period when the duration between any two successive center points is less than 350 milliseconds. This ensures that the ball did not go out of frame for a long time, but also allows for dropped frames with the cameras recording at approximately 10 fps. We then interpolate the center coordinates only for time ranges where both cameras have valid data. Performing this for all valid time ranges gives a set of points for each camera that should correspond to the same location in a global frame. These points determine the transformations, as described in the theory section.

Tests and Results

In the calibration, errors exist due to the propagation of error in determining the ball's center point. We can estimate the uncertainty of the depth measurement in a single camera by looking at the standard deviation of the center location when the ball is stationary. The magnitude of the standard deviation in x, y and z as a vector is plotted against distance to the camera in Figure 8. This plot shows that there is no significant trend in sphere center uncertainty with distance from the camera, but it is always significant, at roughly 3 centimeters on average.

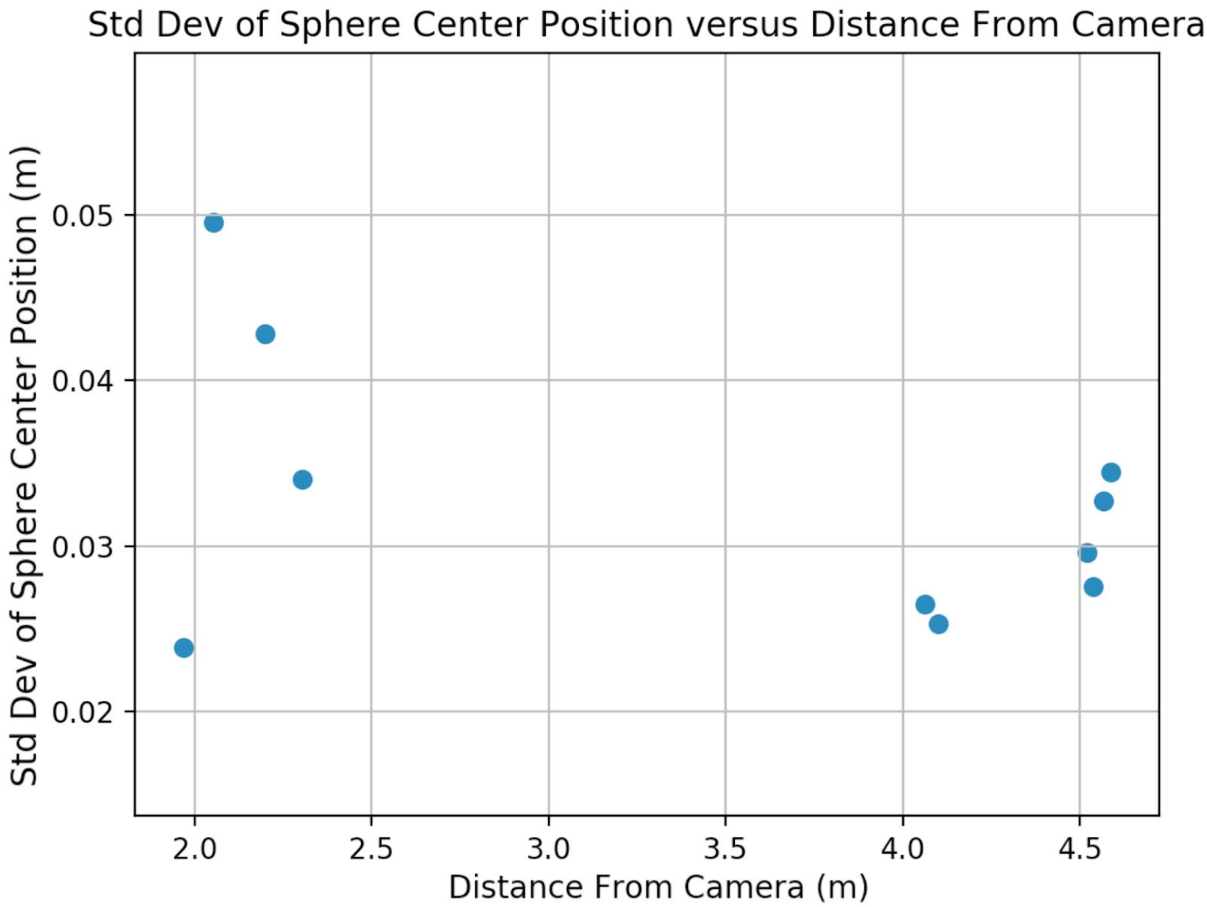

Figure 8. Magnitude of standard deviation of ball center versus distance to camera.

We define the error in the pairwise calibration to be the average distance of transformed points from one camera to points from the other camera. If the ball center is determined exactly in both cameras' coordinate system and the extrinsic transformation is perfect, then this error should be zero. In the case where there is uncertainty on the ball's center in each camera's coordinate system, then this is a lower bound on the error in the calibration. The increase in the error of the calibration from the uncertainty in the ball's center is an estimate of the error in the extrinsic transformation. The average error for mapping cameras 2, 3 and 4 to camera 1's coordinate system for each of the transformation methods is tabulated in Table 1.

Transformation Method	Average Error (m)
Rigid	0.139
Affine	0.116
Non-mixed Second Order	0.105
Mixed Second Order	0.104

Table 1. Average error in extrinsic transformation for each transformation method.

These errors are significantly higher than the 3 cm uncertainty in the sphere center detection, indicating that the transformations are not perfect. This is likely because the camera intrinsics have not been completely corrected by the RealSense software, and the first and second order fits cannot fully adjust for this.

We can also qualitatively evaluate the performance of the extrinsic transformations by looking at merged point clouds, examples of which are shown in Figures 9 and 10 on the following pages.

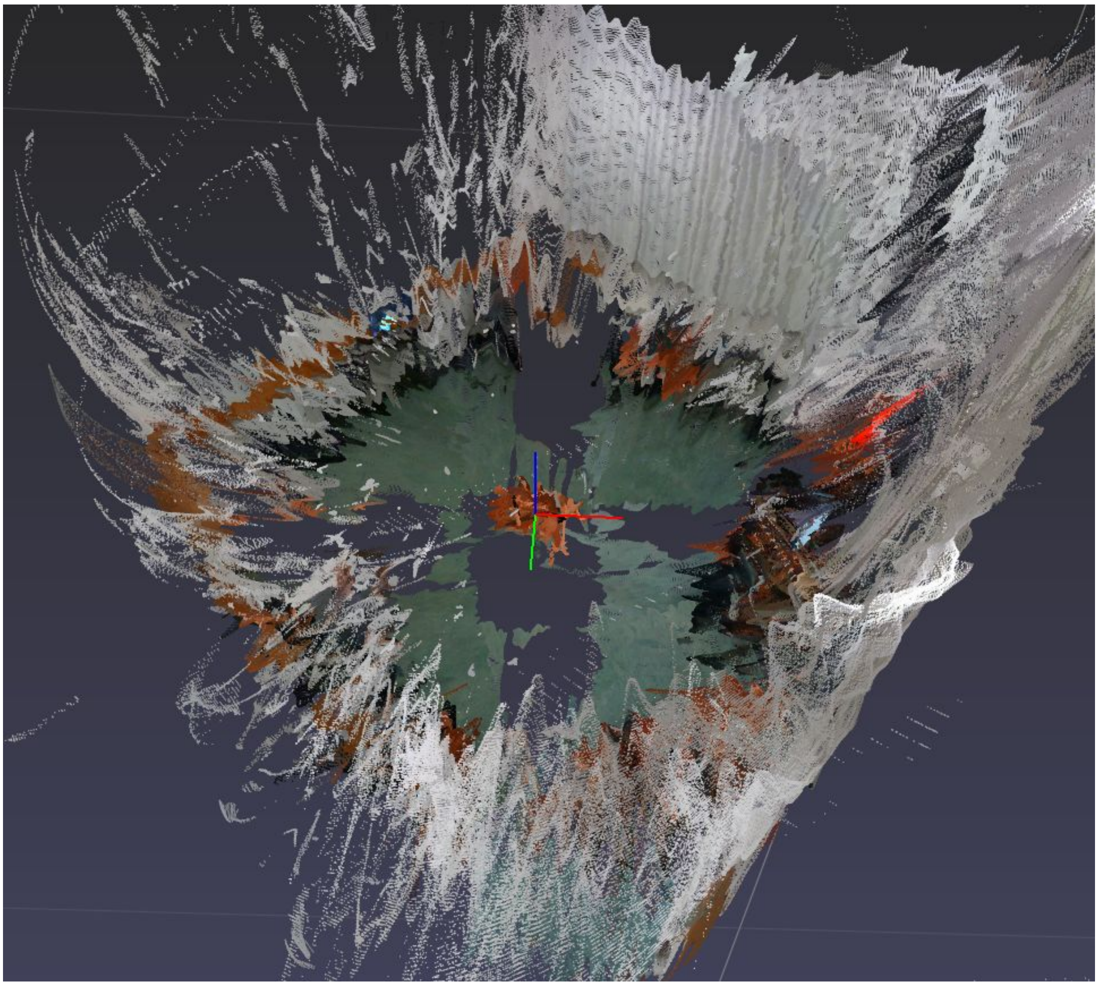

Figure 9. Merged point cloud from all four cameras using mixed second order regression (left), and image of the room from approximately the same point of view (right). A chair is in the middle of the room (orange). The curved patterns around the edges of the room are a result of the quadratic fit. At these points, the calibration is particularly bad, as the calibration path only traverses the center of the room, and the mapping near the wall is an extrapolation.

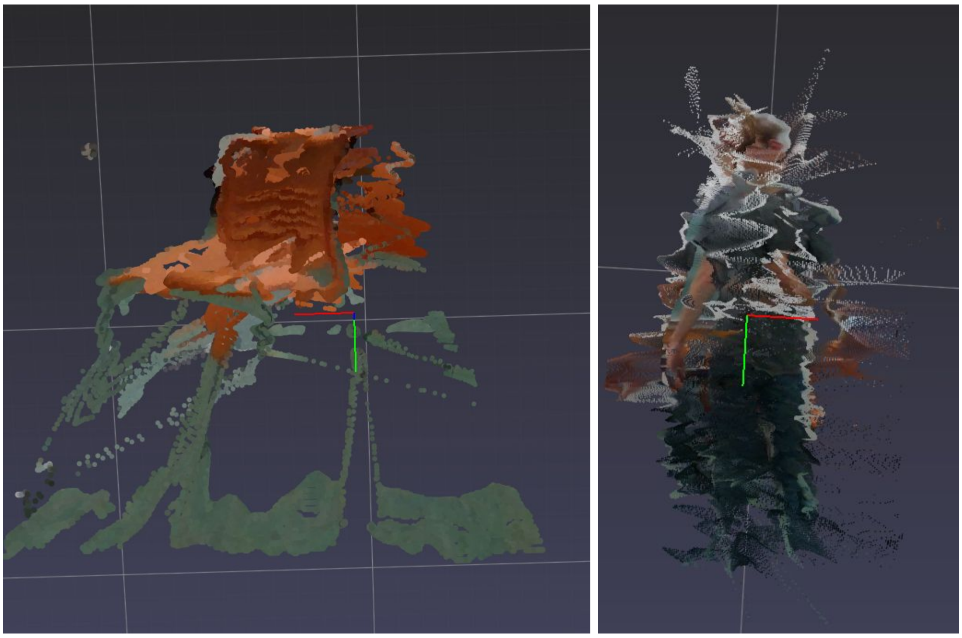

Figure 10. A chair (left) and a person (right) in the middle of the room with surrounding points removed. Point cloud comes from all four cameras calibrated using mixed second order regression.

Figure 10 shows that the extrinsic mapping places the same object in the same rough location, but the result is messy, indicating some error. The wavy patterns and wall colour especially in the image of the person also indicate that there are still errors due to camera intrinsic properties.

Conclusions

In our previous work, we conclude that the lack of an efficient, scalable and accurate skeletal capture system is the key component preventing machine vision powered smart gyms. The goal of this work was to determine if current technology is capable of imaging large areas and generating accurate skeletons of subjects. Based on the results obtained from the Nuitrack software and the Intel RealSense Cameras, we can conclude that this combination of hardware and software is not currently viable for use in a skeletal capture system due to poor accuracy. Despite this, these results do not imply that other combinations of hardware and software would not be viable options for a skeletal capture system. Further tests of different software and hardware systems will be important to verify if any other configuration will suit the requirements of a skeletal capture system for gym automation.

After identifying that the combination of Nuitrack and Intel RealSense Cameras would not be a viable option for a skeletal capture system, we investigated the raw depth data of the Intel RealSense Cameras, while developing an auto-calibration system for multiple cameras. We concluded that even after calibrating these cameras to account for intrinsic issues, many anomalies still exist in in the depth information provided by the cameras. This further supports the conclusion that the RealSense is not a viable sensing system in its current state.

As we were never able to build a working skeleton generation system, we were unable to perform any analysis on the cost per area covered of our system, which was an initial goal of this project. Further work will be required to determine the cost to implement a machine vision-based exercise tracking system in a commercial gym. Building a skeletal generation system will show us how much compute power will be needed to generate a skeleton from a point cloud, and how cameras should be spaced for ideal skeleton generation.

Recommendations

The primary focus for future work should be determining whether the poor performance of the RealSense is due to the hardware itself, or if the data can be improved through more post-processing. It is possible that since we were using new Python APIs to collect data that there were bugs that lead to nonideal data, such as the intrinsics calibration not being properly applied, or the IR emitter not being turned on. It would be worth looking into the more mature C++ API to collect data as bugs are less likely. The pipeline that we have developed could then use saved data to easily evaluate the performance.

If the data from the RealSense Camera improves, then work should shift to generating a skeleton from the merged point cloud. As a first pass, our original suggestion of back-projecting to an ideal point of view to feed through Nuitrack could work, but the act of projection decreases the amount of useful information in the input. Future work should focus on utilizing the extra data that exists in the point cloud, instead of simply a projection to determine a user’s skeleton. Developing a method to generate a skeleton from the point cloud is scalable to any number of cameras and would not explicitly depend on the configuration of the cameras, so we believe this is the optimal solution.

Another direction for future work could be to improve the skeleton generation from a single camera. If this could be done well enough, then our original plan of centrally combining skeletons could be viable. The fact that Nuitrack, a company whose work focuses on skeleton generation, cannot do this well, however, indicates that this is likely a difficult task. The requirement of keeping cost and therefore available computing power as low as possible also makes it more difficult.

Deliverables

The deliverables of this report consist of:

1. A Github repository containing all of our code as well as documentation for running it
2. A hard drive that contains all of the data used to generate the plots and figures in this report
3. The calibration ball

Bibliography

[1] Pernek, Igor and Hummel, Karin Anna and Kokol, Peter (2013) *Exercise Repetition Detection for Resistance Training Based on Smartphones,* Springer-Verlag: London, UK

[2] Li, Chuanjiang, Fei, Minrui et al. (2013). *Free Weight Exercises Recognition Based on Dynamic Time Warping of Acceleration Data*, Springer: Berlin Heidelberg

[3] Sundholm, Mathias, Cheng, Jingyuan (2014). *Smart-mat: Recognizing and counting gym exercises with low-cost resistive pressure sensing matrix,* UbiComp 2014 - Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing

[4] Velloso, Eduardo, Bulling, Andreas (2013). *Qualitative activity recognition of weight lifting exercises,* ACM International Conference Proceeding Series

[5] Hristo Novatchkov and Arnold Baca (2012). *Machine learning methods for the automatic evaluation of exercises on sensor-equipped weight training machines,* Engineering Of Sport Conference

[6] Su, P. C., Shen, J., Xu, W., Cheung, S. S., & Luo, Y. (2018). *A Fast and Robust Extrinsic Calibration for RGB-D Camera Networks. Sensors* (Basel, Switzerland)

[7] Hartley, Richard, Zisserman, Andrew (2003). *Multiple View Geometry in Computer Vision*. Cambridge University Press

[8] Yin-Poole. Wesley, (2017). *Kinect Officially Dead* Eurogamer (London, UK)

[9] Intel Corporation (2018). *Intel® RealSense™ D400 Series Calibration Tools* (Santa Clara, US)

[10] Intel Corporation (2018). *Intel® RealSense™ D400 Series Datasheet* (Santa Clara, US)

[11] Simek, Kyle (2013). *Dissecting the Camera Matrix, Part 3: The Intrinsic Matrix*, Retrieved from http://ksimek.github.io/2013/08/13/intrinsic/

Appendix A - Review of Alternative Solutions

This section will describe alternative approaches to the workout tracking problem and related problems. The following approaches have been taken by for-profit corporations in the United States and Canada. Each analysis will attempt to address a category of solutions as opposed to the solution of a particular corporation. Analysis of each solution will be summarized in the following format:

Description

A brief description of the category of solutions and how they relate to the workout tracking problem.

Corporate Examples

A sampling of corporations who have an implementation of this category of solutions.

Benefits

A description of the favourable attributes of the category and reasons for its continued use in the market today. A successful implementation of our solution should emulate as many of the benefits of existing solutions as possible.

Limitations

A description of what is not feasible with this category of solutions and reasons for which the limitations disqualify this category as an appropriate solution to the workout tracking problem. A successful implementation of our solution should eliminate as many of the limitations of existing solutions as possible.

Appendix A-1 - Manual Pen and Paper Tracking

Description

Users bring a pen and notepad or printed sheet to the gym with them for their workout. After each set of each exercise, the user writes the corresponding weight and number of repetitions on the sheet of paper. These workout records are stored reviewed periodically by the user and stored for future reference.

Corporate Examples

- Bodybuilding.com
- ExRx.net

Benefits

- Does not require technological competence for use
- Little to no privacy risk due to localized data
- Assists the user with motivation and data-driven workout routines

Limitations

- Very input intensive, requires significant effort to record exercises being done
- Very difficult to transpose data to other mediums of presentation
- Disruptive to the user’s workout

Appendix A-2 - Digital Workout Planning and Tracking

Description

Users download a mobile application on their smartphone and bring that with them to their workout. After each set of each exercise, the user types the corresponding weight and number of repetitions into their smart device. These workout records are presented to the user to showcase their progress and facilitate social interaction about their workout routine.

Corporate Examples

- MyFitnessPal
- Bodybuilding.com
- 5x5 StrongLifts

Benefits

- Gives user powerful workout summaries
- Enables workout plan and progress sharing and other social capabilities
- Provides users with continuous feedback and motivation

Limitations

- Very input intensive, requires significant effort to record exercises being done
- Disruptive to the user's workout
- No guarantee that users are reporting accurate information

Appendix A-3 - Wearable Technology

Description

Users purchase a wearable device such as a smart watch. The user wears this device on their person, ensuring it has adequate battery charge for the fitness application. The user then performs their desired fitness activity. Currently, automatically tracked activities are limited to cardiovascular exercises like running, cycling, rowing and elliptical exercises.

Corporate Examples

- Apple Watch
- Fitbit
- Moov
- Atlas
- Garmin

Benefits

- Specific streams of automatically tracked information
- Requires effort to falsify tracked data

Limitations

- Not tailored to activities in fitness centres, tracking of weight training is not feasible with these devices
- Tracked data can be falsified e.g. by attaching the wearable device to a ceiling fan

Appendix A-4 - Equipment Monitoring Sensors

Description

Small accelerometers and other sensors are mounted to pieces of equipment in a fitness centre. This equipment includes treadmills, ellipticals and various weight training machines, but excludes free weights. When gym users interact with the tracked equipment, data is sent to gym owners and managers who use the information to guide their purchasing and maintenance decisions.

Corporate Examples

- Gymtrack

Benefits

- Equipment usage data is valuable to gym owners, most of whom do not currently track this information
- Inexpensive
- Can be standardized to accommodate equipment from any manufacturer

Limitations

- Not capable of tracking free weight exercises, the largest category of weight training exercises, which 76% of gym users perform regularly
- Not able to distinguish users from one another
- Not able to distinguish different exercises on the same machine from one another
- Requires frequent maintenance in the form of battery charging

Appendix A-5 - Equipment Integrated Sensors

Description

Exercise machines come installed with exercise tracking capabilities directly from the manufacturer. Users are able to register their personal profile with these machines. Exercise data is displayed in real-time to the user through visual interfaces installed on the machine. When finished, the user is able to export their personal data to their smartphone via Bluetooth.

Corporate Examples

- eGym
- True Fitness

Benefits

- Equipment usage data can be provided to gym owners and managers
- Visual feedback appeals to users
- Excellent tool for personal training and classes

Limitations

- Not capable of tracking free weight exercises, the largest category of weight training exercises, which 76% of gym users perform regularly
- Prohibitively expensive
- Requires a source of power for every workout machine
- Requires user sign-in on each machine. This is especially inconvenient for users that do circuit-type routines

Appendix B - Bill of Materials

Part	Quantity	Description	Link to purchase
Intel RealSense D415	4	Depth Camera	https://www.mouser.ca/ProductDetail/intel/82635asrcdvkhv/?qs=wd5RIQLrsJjP3SvjhPxxZA==&countrycode=CA&currencycode=CAD
Intel BOXNUC8i7BEH1	2	I7 NUC	https://www.newegg.ca/Product/Product.aspx?item=N82E16856102209
16GB DDR4 SODIMM RAM	2	RAM for NUC	https://www.amazon.ca/Crucial-PC4-19200-SODIMM-260-Pin-Memory/dp/B01BIWMWVS
Samsung 970 EVO 250GB	2	SSD For NUCs	https://www.amazon.ca/Samsung-970-EVO-250GB-MZ-V7E250BW/dp/B07JJ7LJQ2
50 Inch Tripod	4	Tripod for cameras	https://www.amazon.ca/AmazonBasics-50-Inch-Lightweight-Tripod-Bag/dp/B00XI87KV8
5 Meter USB 3 Extension Cable	4	Extension cables for cameras	https://www.amazon.ca/CableCreation-Meters-Extension-Extender-Female/dp/B0179MXKU8
12-inch Styrofoam Sphere	1	To be used with calibration	https://www.amazon.ca/FloraCraft-SFBA12HHU-SmoothFoam-Hollow-White/dp/B01CUTR2BI
Red-Orange Fluorescent Spray Paint	1	For painting calibration ball	https://www.canadiantire.ca/en/pdp/fluorescent-marking-spray-paint-312-g-0482513p.html#srp