Visualization of Carbon Monoxide Particles Released from Firearms

Sadan Suneesh Menon and Thomas Wischgoll; Wright State University; Dayton, Ohio, USA

Abstract

A significant number of soldiers have come forward to report discomfort, irritation, and respiratory problems after taking part in live firing sessions. These problems appear to be caused due to the fumes and particulates emitted from the gun upon firing. There exists substantial research focused on lead and other harmful metallic particulates expelled from a firearm as those are considered the most harmful among the emissions. However, our research focuses on visualizing the carbon monoxide (CO) particles released from a firearm to improve our understanding of their adverse effects on the human body. We use data provided by researchers at Wright-Patterson Air Force Base (WPAFB) enhanced with analyses of provided video material to devise a visualization that shows the correlations between the concentration of CO particles and Lung Deposited Surface Area (LDSA) values as well as other relevant parameters. The results are summarized in the form of a dynamic parallel coordinates plots for in-depth analysis by the domain specialists. Results of this study may be used to glean information about the interrelation between CO particles released and health issues faced by individuals after firing a weapon during a training excercise.

Introduction

Handling a firearm is one of the de facto requirements of being in the armed forces. It is a fundamental necessity for any soldier to learn and practice the use of a firearm. The United States military offers a regulated environment for soldiers to practice the handling and usage of their respective firearms. Hence, soldiers have prolonged exposure to this type of environment, even outside the battlefield. This exposure to particulates and substances released from a firearm causes various health problems in soldiers. We focus our research based on a study conducted in a controlled live fire practice session of soldiers using handguns, i.e. small arms. Airmen have often come forward and reported health issues while firing small arms. Most of the reported issues have been related to respiratory discomfort. Fumes emitted when small arms are fired have been known to cause health problems to the people who come in contact with them during firing. These fumes contain a variety of gaseous and solid particles, but only lead has been extensively studied as a harmful particulate. A study conducted by Voie et al. has shown that even the use of unleaded ammunitions cause acute health problems [26]. The exposure to fumes that are released when firing small arms can even cause long-term health problems to the shooters. While using unleaded ammunition, the fumes emit copper in combination with zinc in high amounts. This may lead to diseases such as inflammation of respiratory organs and metal fume fever [26]. Wingfors et al. have determined that the gases released while firing unleaded ammunition mainly consist of carbon monoxide (CO), carbon dioxide (CO2), ammonia (NH3), hydrogen cyanide (HCN), and formaldehyde. The particulates emitted at the same time contain major concentrations of copper (Cu), zinc (Zn), and iron (Fe) [28]. Even though these metals have proven to not be as dangerous as lead (Pb), they have, however, been associated with various health problems, particularly zinc. It was also observed that particulates that are produced immediately after a gun is fired have diameters that qualify them as being nano-particles. These nano-particles have been known to cause respiratory problems for shooters and are highly toxic. This warrants a thorough examination of the health effects in people who inhale such particles. Therefore, good housekeeping and ventilation should be applied where unleaded ammunition is used to reduce the adversity of the effects caused while firing a gun. The research work of Wingfors et al. also contains accurate figures of the emission factors that may be useful to develop ammunition and propellants that offer less risk to the shooter [28]. The research conducted by Voie et al. [26] has been thorough in analyzing, documenting, and studying the gases, specifically, the particulates copper (Cu) and zinc (Zn) emitted from firearms. Wingfors et al. [28] have provided a detailed account of the emission factors and suggested certain preventive measures. The focus of these papers and several others like them have been on the elements that have been identified as having the most adverse effects on the human body. However, the current study attempts to analyze the carbon monoxide (CO) particles that are released when a gun is fired and to provide a visualization of the correlation between the concentration of CO particles and how it affects the user of a firearm. The knowledge from this study may be useful in determining how harmful or innocuous CO really is while firing a small arm as well as the effectiveness of different ventilation options. The data we used is provided to us by the Wright-Patterson Air Force Base (WPAFB). The concentration of CO and additional data were recorded in an indoor firearm training session.

This data was also accompanied by a fifteen minute long video recording of the live fire session. The video shows four shooters in the ongoing training session, an instructor walks behind them carrying the equipment used to record the particulate data and also aides shooters in the target practice. The equipment used to measure the particle concentration are called Ultrafine Particle Counter and Partector. Partector is used to measure nanoparticles and estimate their respective Lung Deposited Surface Area (LDSA). LDSA is essentially the efficiency with which ultrafine particles (or nano-particles) deposit themselves on to various regions of the lung. It is a measure of surface area and is measured in m2/cm3. In this project LDSA allows us to analyze the ultrafine particles that make it to the lung. We used this data and interpreted the distance between the instructor and the shooters based on the provided video. We then consolidated all this information into single data file and used this data file to create parallel coordinate plots which is the chosen visualization tool to understand the correlations between the data elements. In addition, we use object tracking in the provided videos to glean information that we deemed to be important and combined this information with the data file provided to us by the researchers at WPAFB. We used this to generate a parallel coordinates plot to hypothesize correlations between CO concentration, LDSA, distance between measuring equipment and shooters, and other dimensions in the new data file that may be used to analyze the effect of CO particles released from a firearm on the human body.

Motivation

Health issues have previously been reported by soldiers after live-fire training sessions. Studies have been performed with respect to 3 different types of ammunition (leaded, modified unleaded, and unleaded) wherein volunteers were divided into three groups each using one type of ammunition. Symptoms faced by the shooters were recorded immediately after and a day after the session. It was found that the number of reported symptoms were greater in shooters who used unleaded ammunition when compared to leaded and modified leaded. The fumes released after firing were found to contain in major concentrations hydrogen, nitrogen, carbon dioxide, carbon monoxide, and water [26]. Our research focuses on how the carbon monoxide concentration in these emissions affects the health of the shooters. Carbon monoxide is released from the firearm because of incomplete combustion. The fact that it is colorless and odorless makes it difficult to detect and it is linked to irrevocable harm, and even death. Even though the levels of carbon monoxide (CO) emitted from a firearm are in low concentrations it can still cause long-term health problems if there is regular exposure to it, which soldiers in the military are certain to have. The ASHRAE (American Society for Heating, Refrigerating, and Air-Conditioning Engineers) recommends a maximum indoor exposure of carbon monoxide to be only 9 parts per million (ppm). However, the data we acquired from WPAFB shows a maximum of 12 ppm during a 15-minute session. Regular or even prolonged exposure of this rate can have devastating effects on the health of the people exposed to it.

Carbon monoxide is extremely dangerous mainly because of its affinity towards hemoglobin in the human bloodstream. Once it enters the bloodstream, carbon monoxide (CO) replaces oxygen bound to hemoglobin hence reducing the oxygen that travels to the brain and heart, making it difficult for the organs to function. Carbon monoxide amasses itself in the bloodstream rapidly causing adverse symptoms like nausea, fever, fatigue, headaches, blurry vision, dizzyness, irritability, and confusion. Most of these symptoms were documented in the study conducted by Voie et. al [26] wherein men that were recruited for the tests reported the majority of these symptoms after a firing session. Large increases in the carbon monoxide (CO) levels can also lead to vomiting, damage to the brain, syncope (loss of consciousness), and even death. Routine occupational activities that involve emission of carbon monoxide have known to cause carbon monoxide poisoning. In addition to binding to hemoglobin, carbon monoxide also binds to other proteins, mainly myoglobin, as well as other enzymes. This interferes with oxidative phosphorylation which is responsible for energy productions in all tissues of the human body [26]. The carbon monoxide levels in hemoglobin are measured using a CO-oximetry of samples of blood gas. Carbon monoxide in blood, specifically in hemoglobin present in blood, is referred to as carboxyhemoglobin. Carboxyhemoglobin levels over 25% are considered lethal to the human body. The initial treatment to carbon monoxide poisoning involves the oxygenation of organs in the body. Data provided by our collaborators at Wright Patterson Air Force Base (WPAFB) contains CO data in parts per million (ppm) and its concentration per cubic centimeter (cc). The AFB also provided other data elements such as Lung Deposited Surface Area (LSDA), air flow velocity in the shooting range, and Coefficient of Variation (CoV%). We were also provided with a video recording of the live fire session. We used the given data along with the measured distance between shooters and instructor to develop a visualization system for determining the correlation between different data elements and their effect on the health of the shooter.

Related Work

One component of our work relies on algorithms that provide object tracking in videos. To achieve this, one of the algorithms that we focused on was the Clustering of Static-Adaptive Correspondences for Deformable Object tracking algorithm (CMT). We used CMT to track the instructor within the provided video during a shooting excercise to enrich the given data. This step provides additional data, such as distances between instructor and the shooters. We then used this information along with the given data to show the correlation between this distance, concentration of particles, and also, the air flow using Parallel Coordinates. In this section, we discuss some of the other works that relates to our project.

Particulates and Gases Emitted from Small Arms

Leaded ammunition is proven to cause health problems mainly by increasing lead content in blood. Using leaded ammunition has also been known to contaminate water supplies. Unleaded ammunition was developed to avoid these issues. However, studies have shown that even the use of unleaded ammunition can lead to significant health issues such as nausea, headache, fever, and respiratory problems. Wingfors et al. have determined accurate emission factors for gases and particulate matter released from firing lead-free ammunition [[28]. This information may be used in the development of new ammunition, and also, in the maintenance and ventilation management of indoor shooting ranges. The study shows that among the gases that are released from firing lead-free ammunition, the Carbon Monoxide:Carbon Dioxide ratio has been very high. Voie et al. have also discussed health effects from firing small arms [26]. This study takes into consideration all the particulates and gases produced after firing and reports each of their exposure measurements with respect to three different types of ammunition used, namely: leaded, unleaded, and modified unleaded. Apart from acute health effects the study also analyzes long term effects afflicted on a persons health from exposure to these emissions. Hence, this study may be helpful in developing measures to reduce exposure. However, our research provides a visualization of carbon monoxide particles released from firing small arms with respect to the measurements recorded in an indoor live fire session. This may be used to assist in understanding how the exposure to carbon monoxide that has been released from a fire arm affects the health of a person and also, the possible preventative measures that could be taken to avoid these adverse effects.

Object Tracking in Video

Video tracking is the process of discovering the location of an object in a given video over time. A study conducted by Mishra et al. explains in great detail the different types of video tracking techniques and the key differences between the techniques [17]. There are different algorithms that can be applied to perform object tracking. For example, Kalman Filters can be used to track foreground objects in a video, as demonstrated by Ergezer et al. [6]. They used the average method to perform object tracking in a video recorded by a static camera. Morphological operators to remove noise after background subtraction is performed and the foreground is determined. Weng et al. [27] proposed yet another object tracking technique in video using the Kalman Filter. This technique segments a moving object in a video selected by user and identifies its dominant color. Then, the system model of an adaptive Kalman filter is set by constructing a motion model. The dominant color (in HSV color space) is then used to track the object of interest in subsequent frames. The result is sent back to adaptive Kalman filter whereby the estimation parameters are adjusted by occlusion ratio. OpenCV provides a video tracking mechanism based on contours and edge detection. Dlib is another library that helps perform video tracking in C++. This algorithm breaks down the video into a sequence of images and the path to these sequence of images is provided as command line input argument to the code [13]. The code then performs tracking of a selected object until the last image in the specified path is reached. Zhang et al. propose a tracking algorithm which employs an appearance model based on random projections such that structure of the original image space is preserved [30]. For videos with low frame rate, the tracking process may be taxing on video tracking algorithms due to increased search space, poor motion continuity, background clutter, and appearance variation of target. To overcome this Yuan Li and Haizhou Ai have proposed an approach using cascade particle filter with several stages of importance sampling [14]. Mean shift iterations could also be used to track an object of interest in a video sequence [3]. One of the algorithms that perform video tracking in an adaptive and effective manner is called the Correspondences for Deformable Object Tracking Algorithm, or CMT in short. We ran the CMT algorithm on the given video and started tracking the instructor by selecting a frame in the video. This is usually the first frame in the video. We then click and drag a rectangle around the instructor to place a bounding box. The algorithm then detects certain keypoints within this bounding box. For each subsequent frame, the algorithm tries to find these keypoints again. This is done using an appearance-based approach to locate and track the keypoints from the previous frame by employing a static model composed of the descriptors. Subsequently, a global search is applied to match the keypoints from the previous frame among the candidate keypoints of the current frame by applying a threshold and also the second nearest-neighbor approach on the distance between descriptors. The keypoints that match background descriptors are eliminated. Since these methods are prone to errors, the algorithm implements a novel way of allowing each of these keypoints to vote for an object center. These votes are then clustered and outliers are discarded [19]. CMT works on grey scale images, so the video was converted to grey scale before CMT processing occurs. This approach based on CMT makes tracking the object very adaptive and efficient. Using CMT, we tracked the position of the instructor in the video and obtained the coordinates that correspond to his position in the video in pixels. Since shooters are mostly stationary during the firing session, we manually identify the shooters coordinates in the video beforehand. We use the distance between each shooters' coordinates to manually identify the approximate real-time distance in feet. Each shooter is 4.5 feet apart, so we laterally identify the pixels between each shooter in our video to be approximately 4.5 feet. CMT Tracking is then applied on the instructor and the difference between the instructor's coordinates and the shooters' will provide us the approximate distance between the two in feet. The Particle Swarm Optimization (PSO) algorithm can also be used for precise human body tracking using a skeleton-driven subdivision surface model with significant GPU requirements having been met [18].

Visualization of High Dimensional Data

The increase in the availability of computing resources has also led to the increase in the ability to collect and generate high dimensional datasets. Structures can now be detected in High Dimensional Datasets using the SOM (ESOM) technology [25]. Bertini et al. present a novel technique of using quality metrics to aid in the visual exploration of meaningful patterns in high dimensional data [1]. Clustering is a popular technique used to deal with high dimensional data space. However, this may be rendered ineffective when the dimensions of the dataset is very high (over 1000). Strehl and Ghosh propose a relationship-based approach to solve this by working in an appropriate similarity space in lieu of the original high dimensional data space [23]. However, high dimensional data may cause a clutter of information that may be too difficult to make sense of when visualized. Several techniques may be used to eleviate this problem, such as Dimension Spacing, Dimension Ordering, and Dimension Filtering. Dimension Filtering is the process of removing some of the dimensions in the dataset, which may be a necessary aspect of visualizing very high dimensional datasets. Various popular approaches to dimension filtering (or dimension reduction) exist such as Kohonens Self Organizing Map, Multi-Dimensional Scaling, Principal Component Analysis, etc [24]. Dimensional Ordering is the process used to reveal important relationship information between dimensions in a dataset by ordering them based on their similarities. In Dimension Spacing, the spacing between two adjacent axes/or angles (based on the visualization used) is varied. However, with visualization techniques like parallel coordinates there is equal spacing between axes by default. Using a non-uniform spacing may provide information about dimensions like similarities between adjacent dimensions, and dimension structure [29]. Automated analysis can be performed to extract potentially relevant structures from visualizations [24]. Liu et al. provide a survey of advances in the visualization of high dimensional data over the past 15 years [15]. There are several ways to visualize high dimensional numeric datasets. Some of the more popular ones are as follows: Scatterplots are point projections of data into 2D or 3D space [9]. Heat Maps are a data visualization technique which constitutes of an array of cells which are each colored based on the values of the data. Survey Plots are projections of data in 2D or 3D space and contain rectangular areas representing each dimension of the dataset. Data are represented using a line and the length of the line corresponds to its dimensional value [9]. Parallel Coordinates use parallel axes in lieu of perpendicular axes to represent data in a dataset. Dimensions are represented by a vertical line with the minimum and maximum values scaled to lower and upper boundaries of the line. Polylines connect axes representing dimensional points [9]. Ward et al. discuss the interactive visualization of large multivariate data in their paper [31]. Parallel coordinates have been used by Steed et al. to visualize data related to climate analysis . They used parallel coordinates to glean important associations between the various dimensions in the data. The approach using parallel coordinates was used in research to determine the relationships between climate variables and recognize the important correlations between them in different geographical regions which is an essential part of climate model evaluation [22]. There are various techniques used to obtain further information from datasets. Brushing is one of these operations wherein subsets of data are selected so that they may be hidden, highlighted or marked [16]. In parallel coordinates, the brushing feature is implemented as an adjustable slider vertically placed on one or more axes in the plot and is an effective method for reducing visual clutter. We use this feature later in our study to focus on the correlations relevant to our analysis. Brushing can prove to be very essential while trying to determine how values in certain axes effect the other dimensions of the dataset.

Parallel Coordinates

One of the most common visualization techniques for high dimensional data, that was even considered to be used in this paper is the Scatterplot. In scatterplots, the graph is a plot of two variables (x and y) measured independently and displayed as individual points on a coordinate axis. There is no functional relationship between x and y [7] and because of this reason, scatterplots did not fit our requirements establish correlation between the axes in our data file. Parallel coordinates are widely used in visualization for data with multiple variable quantities and high dimensional geometry. We also considered the use of Streamgraphs. Streamgraphs take a layered approach to visualizing high dimensional data, and is an extension of the stacked area graphs. Streamgraphs are placed around a central axis and hence, have an organic and flowing shape. Byron et al. explain the inception and application of streamgraphs [2]. However, we opted to not use streamgraphs to visualize the results of our project because the results obtained from streamgraphs were not as legible as the ones obtained from parallel coordinates and displaying correlations between axes seemed to work better with parallel coordinates. Parallel coordinates contain axes that are parallel to one another (either horizontally or vertically). A cartesian coordinate system is used along the axes. Lines originating from a point on one axis to a point on its adjacent axis represent the projection of the point to the plane [10]. We used parallel coordinates for this project because they are best suited for multivariate data like ours and can provide precise indication of the correlation between each axis to any other axis on the plot. A color gradient is used to show the change in values of an axis and how this change corresponds to the values of other axes. We also use a brushing tool which is used to click on an axis and create a type of sliding window on that axis to filter out data outside the range of this window. This allows us to focus on the value ranges we want on any of the axes.

System Design

The objective of our project was to visualize particulate data in correlation to the instructor's position. This visualization would be useless to analyze the adverse effects of these particulates on the human body. Our system is divided into 2 parts: Video Tracking and Parallel Coordinates plot. The layout of the system is shown in Figure 1. We discuss these two aspects in depth in the next section.

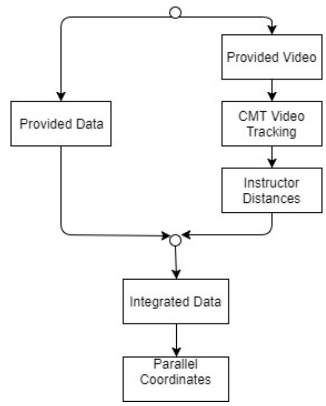

Figure 1. *System Design.*

Researchers at the WPAFB provided us with the data file in Comma Separated Value (.csv) format along with the video recording of the live fire session in MP4 format (any video format can be used in our C++ code). We first tracked the coordinates of the instructor in the given video in order to find the distances between the instructor and the four shooters in the video. We did this by first finding coordinates of the instructor using the CMT video tracking algorithm in the given video. Since we know that the shooters are 4.5 feet from one another we use this information to figure out how far the instructor is from all the four shooters throughout the video. This measure is important to understand how the distance affects the concentration of CO particles measured by the sensor worn by the instructor. To accomplish this, we used a C++ code of the CMT algorithm implemented using the 2013 version of Microsoft Visual Studio. The code uses the OpenCV library and takes the video as a command line argument. The user is prompted to initialize a tracker on the instructor's head so as to record lateral movement of the instructor in the video in the form of coordinates. Since the coordinates of the shooters had previously been identified manually, the tracker's measurement was used to calculate the distance between the instructor and the shooters. This calculated distance (difference between coordinates) was then caliberated by approximating lateral coordinates to identify the actual distance in feet. The approximate distances obtained from using CMTs video tracking algorithm (which uses the OpenCV library) were then added to the existing data file which is in the CSV format. To this data file we added other data elements provided to us by WPAFB such as Air Flow Velocity, Coefficients of Variation(CoV%), etc. Hence, we form a collective data file which contains all the elements that are of interest to us. This integrated csv data file was then used to plot the parallel coordinates which was, in turn, used to observe results and correlations. The parallel coordinate plot is produced using a JavaScript code that implements Data Driven Documents

(D3) library and is executed in the Netbeans IDE. The output is generated within the browser dynamically and allows the user to interact with the visualization. It is compatible with most web browsers (not including Internet Explorer). D3 supports realtively large datasets; it fails at 500000 data points. D3 performs with minimal overhead while also providing interactive visualization and animations. OpenCV is an open source library that facilitates machine learning and computer vision. OpenCV can be used in programs written in Python, Java, C, and C++ languages. It is also availabkle across different platforms. OpenCV has been used for various machine learning and computer vision applications, such as facial recognition software, egomotion (process of approximating a cameras motion with respect to a firm scene), stereopsis (the process of perceiving depth with the help of 2 cameras hence simulation eye sight), motion tracking, as well as various other applications. Even though Matlab programming language also offers image processing tools similar to what was required, we opted to use OpenCV over Matlab because OpenCV tends to be faster compared to Matlab. OpenCV also uses less computer resources than Matlab making it the more efficient choice. Most importantly OpenCV is compatible with the CMT video tracking algorithm which allows us to efficiently perform object tracking in a video. The CMT algorithm and Parallel Coordinate plots are explained in detail in the following section, along with other methods considered to obtain the instructor coordinates in the video.

Implementation

Along with the given data, we were also provided with a 15 minute long video footage of the live-fire training sessions. The given video footage shows four shooters firing at targets in an indoor facility. An instructor walks behind the shooters to assist them with the shooting. The instructor is carrying the equipment used to measure the CO concentration. This equipment used is a Partector, which is a nanoparticle detector, and a Portable Ultrafine Particle Counter (PTRAK Ultrafine). Some of the parameters are known from the session, like airflow velocity at each stall, and that each stall (and therefore each shooter) is 4.5 feet apart from one another. We used the CMT video tracking algorithm in our C++ code to track coordinates of the instructor in the video. We determined that it would be helpful to know how far the instructor was, and hence the measuring equipment is from the shooters at any given point in the duration of the video. This information was then integrated with the given data file. The integrated file was used to plot Parallel Coordinates using a JavaScript-based implementation. This can be used to find out by what factor the recorded CO concentration varies with the distance between the shooters and the instructor. For this project, we experimented with different already existing libraries, such as D3, OpenCV, and Dlib, to identify the most appropriate approach.

Data Driven Documents

Data Driven Documents (D3) is a JavaScript library which allows data to be bound to a Document Object Model (DOM). Data-driven transformations are then applied to the model. D3 lets the user take advantage of modern browsers and provides a plethora of visual interfaces to choose from that are both interactive and dynamic. D3 uses in-built functions in JavaScript for element selection, creating Scalable Vector Graphics (SVG) objects, making them dynamic, styling them and making them interactive. A number of D3.js functions exist that facilitate the creation of rich graphical effects and interactive visualizations. Graphs and charts can be generated using JavaScript functions on SVG objects. These objects can be styled using external style sheets like Cascading Style Sheets (CSS). D3.js uses CSS to select nodes of DOM objects, either individually or collectively, and modifies them. Attributes, styles, and other properties are implemented as data functions in D3.js and not just as constants. Using these properties as functions makes them a powerful tool when implementing more complex features. Debugging D3 is a simple process since it can be accomplished using modern browser's inbuilt inspector. Transformations can be added to attributes where applicable using D3.

Image Processing Libraries

Open Source Computer Vision (OpenCV) library is used in the source code to accommodate image and video processing, and also, for tracking objects in the video. OpenCV is a library that contains over 2500 algorithms to facilitate computer vision and machine learning tasks. OpenCV provides an in-built function to find contours (which are curves joining uninterrupted points on boundaries having the same color) which were not as effective or adaptable when it came to tracking an object as compared to CMT. OpenCVs Color Detection and color-based object tracking also turned out to be ineffective. Dlib [13] is another machine learning library that allows image processing. This algorithm required the video to be stripped down to a sequence of images and the bounding box around the object of interest was placed at the first image and run throughout the sequence. This method was also not accurate enough because the bounding boxs size remained fixed and did not adapt to the change in location of the object of interest in the image sequence.

Approaches Considered

One of the first approaches that was considered for video tracking involved using a track bar to set HSV (Hue Saturation Value) on the video in order to filter out all other colors in the video except one. Then we used OpenCVs contour tracking to track the object of that color. We used background subtraction in the video to remove the static objects in the video and to make only the moving objects prominent. In this way, we created a foreground mask which only contains moving objects that are white while the background is removed, or made black. Background is determined by subtracting the current frame from a background model (previous frame). The objects of interest are narrowed down by using a threshold. This is where we use a HSV track bar which allows us to set a certain HSV value to act as threshold and focus on the moving objects that are within that color range. In the given video, the instructor who is walking behind the shooters is wearing a distinctive red hat while the shooters are either not wearing hats, or are wearing camouflage hats matching their uniforms. Hence, we contemplated that it would be effective to try and focus on the red color and use that as threshold value for the background subtraction. We used OpenCVs contour tracking on this new thresholded image to track the position of the instructor in the video. This approach is shown in Figure 2. However, this method proved ineffective because the bounding box of the tracker kept losing the position of the hat when the shape of the hat changed as he moved around. Hence, we had to consider a different approach.

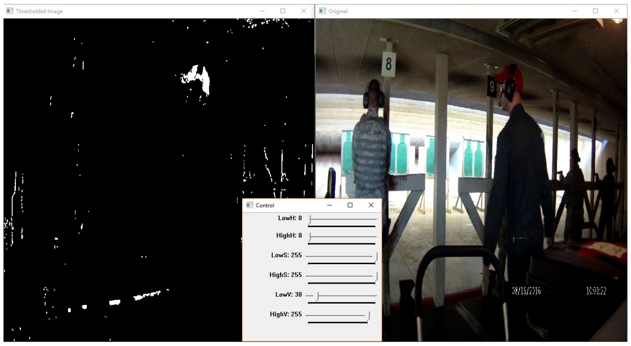

Figure 2. *HSV Thresholding.*

We then looked into Dlib, which is a C++ library that facilitates machine learning operations. Dlib has a video tracking algorithm which requires the video to be broken down to a sequence of images. The set of images are to be put in a directory, the path to which is to be included in the code. A bounding box is to be set on the first image (first frame), and the object selected is tracked until the last image in the directory. The location of the object within the bounding box is identified in the subsequent images. This method was also effective only till the shape of the object being tracked (instructor) changed too much. Also, in the given video the instructor leaves the frame on screen several times. Dlibs video tracking algorithm crashes at this point and requires manual resetting of instructors position and a new first frame for tracking. Dlib also has a substantial overhead with regard to its preprocessing step of converting a given video into a sequence of images before the code is run and using these images as input for the algorithm. These reasons make Dlib less optimal in trying to accomplish what we want from the given data. Hence, we decided not to use this technique and moved to pursue the CMT video tracking method.

Tracking

We looked into several object tracking algorithms to track the instructors coordinates. Some of the algorithms that were considered were tracking with OpenCVs inherent contour drawing, Dlib, and CMT. CMT was chosen in the end for its adaptive nature and accuracy while tracking a moving object (the instructor) in the video. However, the video was not of ideal quality since we were faced with several issues such as low lighting where the shooters are present which makes it difficult for the algorithm to determine depth and identify color differences. Furthermore, the video is recorded with a body cam that is docked on an unstable platform which makes the camera shake throughout the video. These factors influence the accuracy of our results. Even so, we managed to make the best of what we have and gathered the information we wanted from the given video. In CMT, the sequence of images in the video are first converted into grey scale images. After running the code, the video is paused at any moment by pressing any key on the keyboard. The screen then prompts the user to click and drag a rectangle on the area to be tracked. Within this bounding box, the algorithm detects several keypoints based on spatial and photometric constraints. Matches are established between the keypoints in the current frame and the keypoints in the initial frame by using a threshold and also, the second-nearest-neighbor criteria is applied on the distance between the keypoints [19]. A dissimilarity measure is applied to compensate the deformation of the object of interest within the bounding box. The matches are then disambiguated by removing potential keypoints if these keypoints are dissimilar to the set of matches. When the instructor leaves the screen in the video, we added an option where the user can delete the CMT tracker object at the press of a specific key. Then the user can allow the video to play on without the tracker until any key on the keyboard is pressed, wherein the CMT tracker object is re-initiated and the user is prompted to set the bounding box again using the mouse. The coordinates of the bounding box rectangle and the width and height of the rectangle were used to find the coordinates of the center of the rectangle which is considered to be the position of the instructor in the video. We also find the positions of the shooters, who are mostly stationary throughout the duration of the video. Since we know that the shooters are 4.5 feet apart from each other as per the configuration of the shooting range, we use this information along with the coordinates of the instructor (as well as the coordinates of the shooters) to estimate how far the instructor is from each of the shooters. We use the distance between the position of the instructor and the position of the shooters to find out the approximate distance between the instructor and every shooter throughout the session in the video. These distances are recorded and integrated into the given CSV data file, and the minimum value of the distance between the instructor and the shooters was also calculated. We used this integrated data file, which also has the airflow velocity for each stall in the firing range, to plot a parallel coordinate graph in order to help us understand the correlation between the CO concentration, air flow, and the distance of the instructor from the shooters. Screen shot of the tracking process of the given video with the bounding box on the instructor is shown in Figure 3. OpenCV contains an in-built class called Mat which may be used as a container for images wherein the pixels of the image are stored in a matrix. Each frame of the video is stored as a Mat object (example: Mat image). The video is input to the code as a command line argument. In the code, we first created a CMT object. We then converted the current frame image into grayscale. We did this by first converting the image into a HSV (Hue, Saturation, and Value) image and then, in turn, converted this HSV image into grayscale. The grayscale image formed was stored in yet another Mat variable, which was used to process our CMT operations until we display the output image on screen when we convert it back to a regular image. At this point the user was prompted to press any key in order to pause the video. Once the video was paused, the control was then passed to the user and prompted to click on the top left corner of the object of interest. After that, the user had to click on the bottom right corner of the object. This formed the bounding box rectangle. we determined the center of that rectangle and logged it into a file for every frame. This served as the coordinate for the position of the instructor within our video. The CMT algorithm attempts to track similar objects found in that bounding box in subsequent frames resulting in the tracking of the instructor. The tracking operation can be terminated at any moment by pressing the key q. If the tracker (bounding box) happened to lose the instructor at any point in the video, we added a functionality to delete the CMT object at any point by pressing the key t. The tracker could be initiated again and the user was then prompted to create a bounding box about the object of interest (the instructor). This was also applicable for the moment in which the instructor leaves the frame of the camera and comes back into frame after a while. Once the coordinates for the centers of the bounding boxes for each frame throughout the video were identified, it represented the position of the instructor in the video. We also knew that each shooter is 4.5 feet away from one another, and the shooters were stationary in the video and did not move away from their positions. Therefore, we found the coordinates of each of the shooter using the tracker beforehand. Using this information, we found the approximate distance between the instructor and the shooters by considering the difference between the coordinates of the instructor and the shooters for each iteration of the tracking process in the video. The distance between the pixels is laterally translated to real time distance in feet. We added this distance information to the given data file so as to observe the interrelation between the CO concentration and the distance between the shooters and the instructor during the live fire session. Because there were four shooters visible in the video, we have shown the distances between the instructor and each of these four shooters as four separate axes in our parallel coordinates plot. The implementation on CMTs algorithm for object tracking in a video revolves around distinguishing the inlier keypoints from the outliers. The algorithm is adaptable in that it propagates inliers even if the potential keypoints lie on obscure regions of the object of interest, i.e. the object within the bounding box.

Figure 3. *CMT Video Tracking.*

Plotting Parallel Coordinates

The development of parallel coordinates, as an idea, began as early as in the year 1978. The idea was conceived in order to yield graphical representation of multidimensional relations [12]. In our project, the integrated data file now contains the time stamp, CO concentration (in cubic centimeter and parts per million), Lung Deposited Surface Area (LDSA), High Voltage (HV), Transmission Electron Microscopy (TEM), Transmission Electron Microscopy - Voltage (TEM-V), Temperature (T), Relative Humidity (RH), Charger Diffusion Current (Idiff), Battery (Bat), flow values, error, instructor distances, air flow velocities, and Coefficient of Variation (CoV%). The data file is in the Comma Separated Values (CSV) format. To this data file we added the distances from the instructor to all the shooters that we measured previously using the video tracker. Since we also noted the exact times in which the shooting takes place from the video, we used the precise time stamp at which firing took place and set up a Shooting Time (ST) dimension. This dimension is set to the value 1 on the exact time when the shooting occurs and 0 when there is no shooting. After the gun is fired, this value linearly decreases to 0 by 10% each second using the formula $ST = 1-(0.1*(ST-1))$. Each of the records in ST is multiplied with the values in the distance column for each shooter (distances between instructor and each shooter) to find the adjustment variable to the distance in terms of when the shooting actually occurs. These new dimensions are labeled Distances with respect to Shooting Time (Dist.wrt ST). All of these additional columns are added to the given data file to obtain an integrated data file in the .csv format. The objective is to plot parallel coordinates for this integrated data set. Parallel coordinates are used to visualize data in our project because it supports data files with a large number of records and facilitates observation of the correlation among the data records. Multidimensional data led to the inception of parallel coordinates. The structure of parallel coordinates is as follows: A number of lines (axes) that is equal to the number of dimensions (each line representing a dimension) are placed perpendicular to the X-axis and equidistant from one another on a Euclidian plane. Each of these axes were labeled respective to the name of the dimension [21]. The relationship between values in each axis were shown by a line drawn between the axes. Geometry representing numerous domains can be visualized using parallel coordinates. Here, a domain can be a function defined for a set of values. In parallel coordinates, axes are placed parallel to one another. The orientation of these axes can either be horizontal with respect to Cartesian Coordinate plane (parallel to x axis) or vertical (parallel to y axis) which is the more commonly used implementation with respect to orientation. Using point line duality to N points for every adjacent pair of axes produces $N-1$ lines, therefore, $N-1$ line segments. Each of these lines illustrates the projection of a point onto the respective plane. Hence, polylines are formed that intersect all axes at the corresponding coordinates [10]. Parallel coordinates are powerful for understanding datasets that are multidimensional. In parallel coordinates, the axes are not placed orthogonally, but instead are parallel to one another (hence the name). It is very easy to investigate the relationship between results, and even a combination of results in a parallel coordinate plot. However, because of the nature of parallel coordinates, one of the major problems is over-plotting which occurs when there is a large multivariate dataset to plot. This can however be improved using a color scheme to represent a dimension in the data. If the data are still large then a simple preprocessing step could be applied, such as clustering. This is used to combine related data, so it seems as one record before the clustered data set is given as input to create the respective parallel coordinate plot. The order of the axes determines what patterns emerge between adjacent axes. This is facilitated in our JavaScript code by allowing the user to click and drag individual axes to any position in the plot. This way an axis can be made adjacent to every other axes in the parallel coordinate plot. It is important to note that the correlation between axes becomes less clear as the distance between axes increases. The advantages and disadvantages of parallel coordinates are discussed in detail by Cuzzocrea et al. [4]. Since we work on a multivariate dataset with the objective to find correlation between each of the dimensions we chose parallel coordinates to be the visual representation of the same. We used Data Driven Documents (D3) which is a JavaScript library used to visualize data. The data file was fed into the JavaScript code which then produced the parallel coordinates. The code was designed to create the graph with certain features such as a color gradient being applied on the CO concentration column of the data so as to make the correlation between columns easy to discern. A color is applied to a computed static value of each row. One dimension (or column) is chosen to be what the color scheme or gradient is based on. In our data, we have chosen the Carbon Monoxide Concentration (CONC.[#/cc]) dimension to be what the parallel coordinate plot is colored by. The code also allows the graph to be reorderable, in that the columns in the graph can be moved to different positions. This can be useful to better understand the relationship between multiple columns by placing them next to one another. Also supported is a brushing feature that allows the user to click and drag a rectangle on multiple columns in order to filter data by selecting a subset of sample points from axes. This way, the user can view the selected values in the data versus the entire data and observe how they correlate with the rest of the data. Upon running the code, the user can also move the dimensional axes around to make better sense of correlations between axes by placing axes of interest adjacent to one another. The parallel coordinate plot is displayed in the default web browser. The parallel coordinates plot produced from the integrated data file when the CONC.[#/cc] axes is taken as reference for the color gradient is as shown in Figure 4. We use brushing, axes reordering, and changing color scheme features on different axes of this plot to observe the interrelations between the data elements in our data file. These interrelations may be used to understand the effect of carbon monoxide emissions from firearms and take the necessary precautionary measures to avoid health problems.

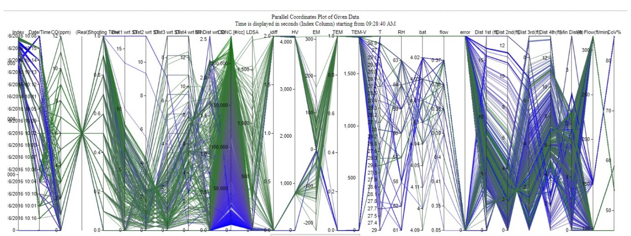

Figure 4. *Parallel Coordinates plot.*

The parallel coordinate plot shows the relationship between each of the dimensions in the dataset. The color scheme is based on the carbon monoxide (CO) concentration since it is of primary importance to us. Hence, the parallel coordinate plot shows how concentration varies with respect to each of the other dimensions such as time, distance of instructor (measuring equipment) from shooter, LDSA, high voltage, temperature, humidity, exact shooting time stamp, and air flow velocity. Figure 4 shows the initial output of the parallel coordinate plot without any user interactions like brushing or reordering of dimensions performed. The correlation of CO concentration is already discernible with respect to other columns. This culmination of information gained using the integrated dataset may be valuable in the interpretation of CO toxicity levels during firing of small arms. The correlations between axes that are deemed important are discussed in detail in the next chapter.

Discussion

Particulates that are less than 100 nanometer (nm) in size or diameter are called ultrafine particles. Ultrafine particles have known to cause health issues of the adverse nature such as cardiovascular diseases and respiratory problems [11]. Oberdorester et al. [20] have surmised that a particles surface area can invoke cellular responses. Higher surface area per mass can make ultrafine particles more biologically active, when compared to larger particles of the same chemistry. Therefore, surface area is a better metric for measurement of particulate matter than mass. However, in the case of health issues, especially of the respiratory kind, only particles that make it to the lung are really relevant. For this, we use a special measure called Lung Deposited Surface Area (LDSA) which considers the efficiency of deposition of nanoparticles within various regions of the lung. Geiss et. al [8] explain in detail how LDSA is measured and how they vary depending on the surrounding environment. We use the parallel coordinates plotted using the integrated data to discern correlations between CO concentration and LDSA. We also observe the correlation between the concentrations and the instructor distances using these parallel coordinate plots. Several other interrelations with regard to concentration, temperature, air flow velocities, and distance from instructor with respect to shooting times, Transmission Electron Microscopy and Coefficient of Variation may be observed using these plots.

When we set the color scheme to be based on the CO mass in ppm we observed by brushing over the CO(ppm) axes that LDSA value is 0 (does not exist) for particles over 4 ppm, as shown in Figure 5. However, when the CO value is between 1 and 4 ppm, we see the LDSA values peak as high as 2000m2/cm3. This shown in Figure 6.

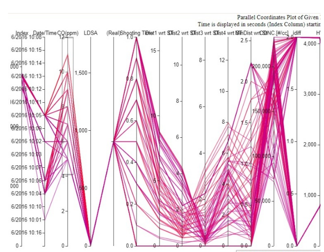

Figure 5. *LDSA when CO is above 4 ppm.*

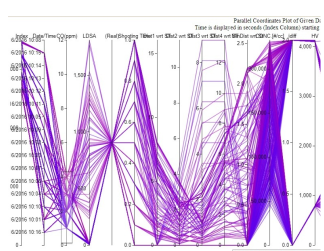

Figure 6. *LDSA when CO is between 1 and 4 ppm.*

Many correlations can be inferred from the parallel coordinates plot in our study. For instance, at first glance it can be determined that the concentration of CO per cubic centimeter is high when the distances between the instructor and shooters are low. It can also be observed that the concentration is high when shooting occurs (as a factor of the Shooting Time dimension). It can also be quickly inferred from the parallel coordinate plot where the color scheme is based on the CO(ppm) axes, that a lesser volume of CO particles is more prevalent during the session as shown in Figure 7. More precisely, the concentration of CO particles is high from 0 to 4 ppm. This can be observed in Figure 8. Since the color scheme is applied on the CO(ppm), the higher volumes of CO particles (in ppm) are shown by red and the color gradient transitions to dark blue as the volume decreases. These observations can be immediately noticed because of the color scheme applied because it shows more blue polylines than red.

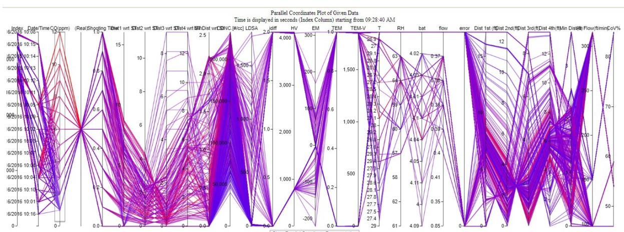

Figure 7. *CO(ppm) Correlation.*

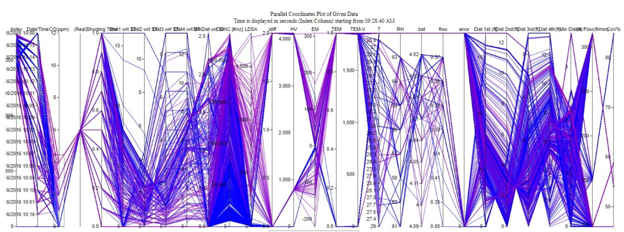

Figure 8. *CO more prevalent between 0 to 4 ppm.*

When the time column is brushed to time at which shooting actually takes place, there is an apparent and significant rise in the CO concentration as shown in Figure 9. Alternatively, when the brushing is one at the times at which there is no shooting, we see a decreased concentration of CO particles. This is illustrated in Figure 10. This shows that the ventilation system at the shooting range constantly removes particulates and gases in the air.

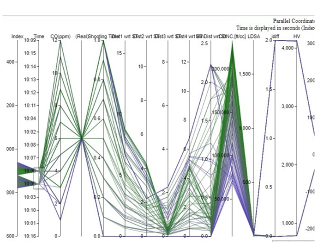

Figure 9. *Concentration of CO When Shooting Occurs.*

The correlation between the minimum distance between instructor and shooter and the concentration of carbon monoxide can also be seen by brushing on the Min Dist wrt ST column. Here, the color scheme is based on the CONC.[#/cc] column. We can observe the Minimum Distance column which is a function of the time at which shooting occurs (Min Dist wrt ST) that the

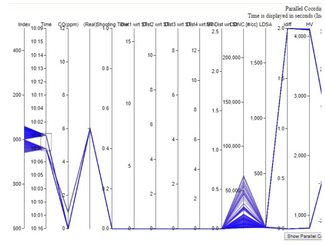

Figure 10. *Concentration of CO When No Shooting Occurs.*

concentration of CO is high when the distance is low and the concentration is low when the minimum distance is high. This can be seen in Figures 11 and 12 respectively.

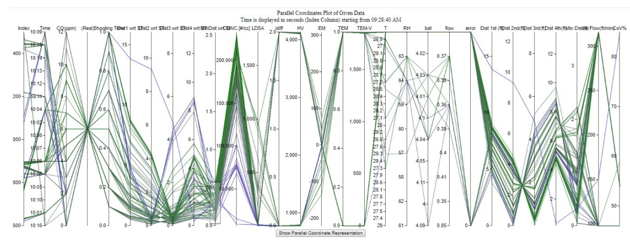

Figure 11. *Concentration of CO When Minimum distance is Low.*

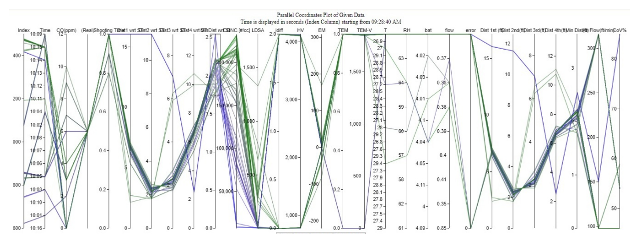

Figure 12. *Concentration of CO When Minimum distance is High.*

By brushing on both CO concentration (#/cc) and LDSA columns, we are able to observe the correlation between the two columns. We see that the LDSA values are higher or more prominent when the concentration per cubic centimeter value is high (between 150,000 and 250,000). This relation is shown in Figure 13. Alternatively, there is a significant decrease of LDSA values for concentration below 150,000 cc of CO, as shown in Figure 14.

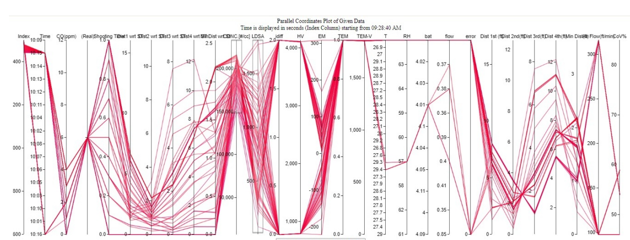

Figure 13. *Correlation Between Concentration of CO per cc and LDSA When Concentration is High.*

The correlation between the concentration of CO particles and the distance between shooters and instructor can also immediately be perceived as shown in Figure 15. The color scheme here

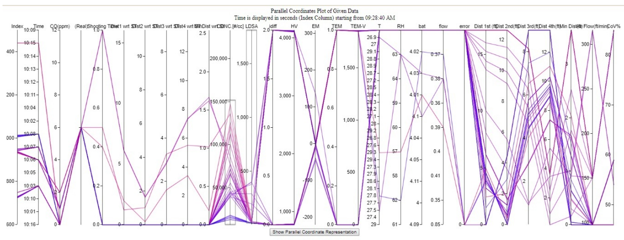

Figure 14. *Correlation Between Concentration of CO per cc and LDSA When Concentration is Below 150,000 cc.*

is based on the CONC[#/cc] column. We use the color scheme and brushing properties of parallel coordinates again to recognize interrelations among these axes. Since we see a plethora of green lines on the Distance columns when shooting occurs it can be concluded that the concentration of CO particles is high when shooting occurs. Since the high concentration lines seem to intersect the distance columns at their lower values, we can also state that the concentration is high when the distance is low. This can be observed in Figure 6.12 and Figure 6.13, where the minimum distance column is brushed along with the concentration column.

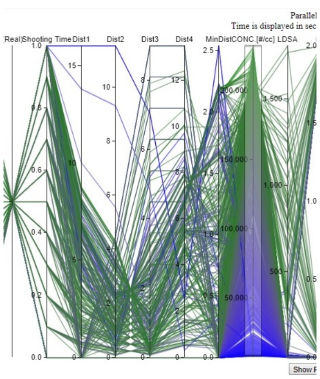

Figure 15. *Correlation Between Concentration of CO per cc and Distance with Respect To Shooting Times.*

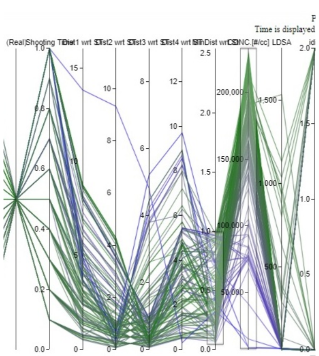

Figure 16. *Concentration of CO is High When Minimum Distance is Low.*

This section has surmised the results of our project. The parallel coordinates plot can be further used to discern correlations between other axes in the data file. New axes can also be added to the integrated data file, if needed. There is also the flexibility to add an entirely new data file to our code. These results

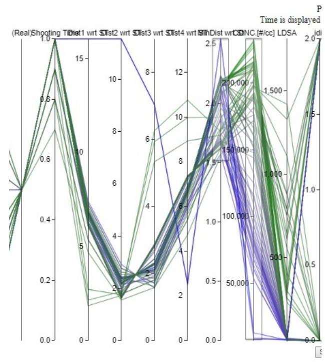

Figure 17. *Concentration of CO is Low When Minimum Distance is High.*

may be helpful to understand and deal with the carbon monoxide emissions from a fire arm. The video processing implemented in this project also gives a more detailed perspective with respect to how far the measuring equipment is when the shooting takes place. However, this analysis of the video could have been made more accurate if two cameras were used to record the live fire session instead of one. OpenCV accommodates mapping depth using multiple images. This process retrieves information about depth intuitively using two different images of the same scene hence imitating stereopsis. By knowing the distance between the two cameras and the focal length of the cameras, the depth of all pixels in the image can be derived. Therefore, this measurement could have been used to find the precise distance of the particle measuring equipment (hence the instructor) throughout the video.

Conclusion

The objective of this project was to assist researchers and scientists in understanding the concerns with possible carbon monoxide (CO) poisoning that may occur after firing a weapon. The video tracking algorithm may be used to track any object in a video of any format, with the option of canceling or removing the tracker at any point in the video. The code for plotting parallel coordinates can also be used with other data files of the csv format, although it is advisable to alter the color scheme section of the code with respect to the axes of interest. In our study, we used the CMT video tracking algorithm to track the instructors coordinates in order to find the distance between CO concentration and LDSA measuring equipment from the shooters at any given time during the live fire session. This additional information was then combined with data provided by our collaborators at WPAFB to analyze correlations between CO concentrations, LDSA, and distance between measuring equipment and shooters using parallel coordinates. Several other correlations can also be analyzed using our visualization based on a parallel coordinates plot. However, we focused on the correlations between the dimensions mentioned above as these are the most relevant ones with regard to our analysis of how CO emissions affect the human body. The parallel coordinates plot in this project has revealed that LDSA is only significant when the parts per million count of CO is low. This means that the particles that cause respiratory illnesses are more prominent in lower concentrations of those particles. Lower concentration of CO are more prevalent after a firearm is shot. Hence, LDSA values are considerably important to note. However, CO concentration picked up by the measuring equipment is relative to distance between the equipment and the shooter. Therefore, this must be taken into account while deriving inferences from this report that may be used to avoid health issues due to CO produced from firearms. LDSA values are observed to be substantially high when CO concentration per cubic centimeter is high. In the data given to us, the released CO particles are over the recommended concentration of 9 ppm in the indoor shooting range. Hence, it is recommended that either a different type of ammunition be developed that reduces CO emission when fired to nullify or minimize the adverse health effects caused by the emissions or more effective ventilation systems be used. Our collaboraters at WPAFB were very excited to see their data enhanced using the trcking algorithm and then represented in the parallel coordinate plots as it allowed them easier access to their data and a way to interactively explore it to find additional correlations. The system presented in this paper may be used to aid in understanding and dealing with the numerous health problems cause due to the exposure to particles emitted from a firearm.

Future Work

In this project, we have focused on attempting to find the correlations between CO concentration, LDSA values, and distance of shooter from equipment in order to aid people to understand how to prevent CO monoxide poisoning and other health issues caused by firing a small arm. The findings from this study may also be used to examine correlations between other data elements by adding new axes to the existing data file or by replacing the data file entirely. This project also focused on object tracking in video which is accomplished using the CMT algorithm. This was used to find the approximate distance of the CO and LDSA measuring equipment from the shooters in the given video. This calculation can be made more accurate if the depth of the images in the video could be mapped out. This can be accomplished by using two cameras at a set distance from each other in order to record the video of the live fire session. Using the distance between the cameras and the focal length of the camera lenses, OpenCV can be used to calculate the depth of the images so that the precise distance between the measuring equipment and the shooters can be determined.

Acknowledgments

The authors would like to thank the Henry Jackson Foundation for supporting this project. Special thanks also goes to Christin Grabinski and Wright Patterson Air Force Base for providing the data set used for this project. This paper was cleared for public release (88ABW-2019-2532).

References

[1] Enrico Bertini, Andrada Tatu, and Daniel Keim. Quality metrics in high-dimensional data visualization: An overview and systematization. IEEE Transactions on Visualization and Computer Graphics, 17(12):22032212, 2011.

[2] Lee Byron and Martin Wattenberg. Stacked graphsgeometry & aesthetics. IEEE transactions on visualization and computer graphics, 14(6), 2008.

[3] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Real-time tracking of nonrigid objects using mean shift. In Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on, volume 2, pages 142149. IEEE, 2000.

[4] A. Cuzzocrea and D. Zall. Parallel coordinates technique in visual data mining: Advantages, disadvantages and combinations. pages 278284, July 2013.

[5] Laura C. Politte Gina N. Duncan. Cristina Cusin Christopher J. Hopwood Davin K. Quinn, Shunda M. McGahee and Theodore A. Stern. Complications of carbon monoxide poisoning: A case discussion and review of the literature. Primary Care Companion to The Journal of Clinical Psychiatry, pages 7179, 2009.

[6] Hamza Ergezer and Kemal Leblebicioglu. Visual detection and tracking of moving objects. In Signal Processing and Communications Applications, 2007. SIU 2007. IEEE 15th, pages 14. IEEE, 2007.

[7] Michael Friendly and Daniel Denis. The early origins and development of the scatterplot. Journal of the History of the Behavioral Sciences, 41(2):103130, 2005.

[8] Otmar Geiss, Ivana Bianchi, and Josefa Barrero-Moreno. Lung-deposited surface area concentration measurements in selected occupational and non-occupational environments. Journal of Aerosol Science, 96:24 37, 2016.

[9] Georges Grinstein, Marjan Trutschl, and Urska Cvek. High-dimensional visualiza- tions. In Proceedings of the Visual Data Mining Workshop, KDD, 2001.

[10] Julian Heinrich and Daniel Weiskopf. State of the art of parallel coordinates. In Eurographics 2013-State of the Art Reports, pages 95116. The Eurographics Association, 2013.

[11] Gerard Hoek, Hanna Boogaard, Anne Knol, Jeroen de Hartog, Pauline Slottje, Jon G. Ayres, Paul Borm, Bert Brunekreef, Ken Donaldson, Francesco Forastiere, Stephen Holgate, Wolfgang G. Kreyling, Benoit Nemery, Juha Pekkanen, Vicki Stone, H.- Erich Wichmann, and Jeroen van der Sluijs. Concentration response functions for ultrafine particles and all-cause mortality and hospital admissions: Results of a european expert panel elicitation. Environmental Science & Technology, 44(1):476482, 2010. PMID: 19958027.

[12] Alfred Inselberg and Bernard Dimsdale. Parallel coordinates for visualizing multidimensional geometry. In Computer Graphics 1987, pages 2544. Springer, 1987.

[13] Davis E King. Dlib-ml: A machine learning toolkit. Journal of Machine Learning Research, 10(Jul):17551758, 2009.

[14] Yuan Li, Haizhou Ai, Takayoshi Yamashita, Shihong Lao, and Masato Kawade. Tracking in low frame rate video: A cascade particle filter with discriminative observers of different life spans. IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(10):17281740, 2008.

[15] Shusen Liu, Dan Maljovec, Bei Wang, Peer-Timo Bremer, and Valerio Pascucci. Visualizing high-dimensional data: Advances in the past decade. IEEE transactions on visualization and computer graphics, 23(3):12491268, 2017.

[16] Allen R Martin and Matthew O Ward. High dimensional brushing for interactive exploration of multivariate data. In Proceedings of the 6th Conference on Visualization95, page 271. IEEE Computer Society, 1995.

[17] P. K. Mishra and G. P. Saroha. A study on video surveillance system for object detection and tracking. pages 221226, March 2016.

[18] Luca Mussi, Spela Ivekovic, and Stefano Cagnoni. Markerless articulated human body tracking from multi-view video with gpu-pso. In ICES, pages 97108. Springer, 2010.

[19] Georg Nebehay and Roman Pflugfelder. Clustering of Static-Adaptive correspondences for deformable object tracking. In Com-

puter Vision and Pattern Recognition. IEEE, June 2015.

[20] Gunter Oberd orster, Eva Oberd orster, and Jan Oberd orster. Nanotoxicology: an emerging discipline evolving from studies of ultrafine particles. Environmental health perspectives, 113(7):823, 2005.

[21] S Savoska and S Loskovska. Parallel coordinates as tool of exploratory data analysis. In 17th Telecommunications Forum TELFOR, Belgrade, Serbia, pages 2426, 2009.

[22] Chad A Steed, Galen Shipman, Peter Thornton, Daniel Ricciuto, David Erickson, and Marcia Branstetter. Practical application of parallel coordinates for climate model analysis. Procedia Computer Science, 9:877886, 2012

[23] Alexander Strehl and Joydeep Ghosh. Relationship-based clustering and visualization for high-dimensional data mining. INFORMS Journal on Computing, 15(2):208230, 2003.

[24] Andrada Tatu, Georgia Albuquerque, Martin Eisemann, Jorn Schneidewind, Holger Theisel, Marcus Magnork, and Daniel Keim. Combining automated analysis and visualization techniques for effective exploration of high-dimensional data. In Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on, pages 59 66. IEEE, 2009.

[25] Alfred Ultsch. Maps for the visualization of high-dimensional data spaces. In Proc. Workshop on Self organizing Maps, pages 225230, 2003.

[26] yvind Voie, Anne-Katrine Borander, Liv Ingunn Bjoner Sikkeland, Svein Grahnstedt, Arnt Johnsen, Tor Erik Danielsen, Kjetil Longva, and Johny Kongerud. Health effects after firing small arms comparing leaded and unleaded ammunition. Inhalation toxicology, 26(14):873879, 2014.

[27] Shiuh-Ku Weng, Chung-Ming Kuo, and Shu-Kang Tu. Video object tracking using adaptive kalman filter. Journal of Visual Communication and Image Representation, 17(6):11901208, 2006.

[28] H. Wingfors, K. Svensson, L. Hgglund, S. Hedenstierna, and R. Magnusson. Emission factors for gases and particle-bound substances produced by firing lead-free small-caliber ammunition. Journal of Occupational and Environmental Hygiene, 11(5):282291, 2014. PMID: 24188168.

[29] Jing Yang, Wei Peng, Matthew O Ward, and Elke A Rundensteiner. Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets. In Information Visualization, 2003. INFOVIS 2003. IEEE Symposium on, pages 105112. IEEE, 2003.

[30] Kaihua Zhang, Lei Zhang, and Ming-Hsuan Yang. Real-time compressive tracking. In European conference on computer vision, pages 864877. Springer, 2012.

[31] Fua, Ying-Huey, Matthew O. Ward, and Elke A. Rundensteiner. "Hierarchical parallel coordinates for exploration of large datasets." In Proceedings of the conference on Visualization'99: celebrating ten years, pp. 43-50. IEEE Computer Society Press, 1999.

Author Biography

Sadan Suneesh Menon *received his bachelors degree in Computer Science and Engineering Sambhram Institute of Technology, Bangalore - India in 2015. He received his masters degree in Computer Science at Wight State University, Dayton, Ohio - USA in 2017. He is currently pursuing his PhD at Wright State University focusing on visualization and data analysis.*

Thomas Wischgoll *received his Master's degree in computer science in 1998 from the University of Kaiserslautern, Germany, and his PhD from the same institution in 2002. He was working as a post-doctoral researcher at the University of California, Irvine until 2005 and is currently a Professor and the Director of Visualization Research at Wright State University. His research interests include large-scale visualization, flow and scientific visualization, as well as biomedical imaging and visualization. In the area of vector field visualization, Dr. Wischgoll completed the topological analysis of vector fields by developing an algorithm that detects closed streamlines, a missing link between branches of a topological skeleton. In the realm of biomedical engineering, he developed a visualization system that facilitates the analyses of large-scale vascular models of a heart represented geometrically by several hundred million polygons. The models are derived from CT scans and feature a simulated flow inside the blood vessels. Dr. Wischgoll developed methodologies for analyzing such volumetric data and extracting quantitative measurements at very high accuracy for further analysis. His research work in the field of large-scale, scientific visualization and analysis resulted in more than thirty peer-reviewed publications, including IEEE and ACM. Dr. Wischgoll is a member of ACM SIGGRAPH, IEEE Visualization & Graphics Technical Committee, and the IEEE Compute Society.*