Efficient Methods for Video-based Human Activity Analysis

By

Xuan Wang

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy
(Electrical Engineering)

at the
UNIVERSITY OF WISCONSIN-MADISON
2019

Date of final oral examination: 8/13/2019

The dissertation is approved by the following members of the Final Oral Committee:
Yu Hen Hu, Professor, Electrical and Computer Engineering
Robert Radwin, Professor, Industrial and Systems Engineering
Bill Sethares, Professor, Electrical and Computer Engineering
John A. Gubner, Professor, Electrical and Computer Engineering
Vikas Singh, Professor, Biostatistics and Medical Informatics, Computer Sciences

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

ACKNOWLEDGMENT

This dissertation would not have been possible without the help from numerous individuals.

I'd like to express my deepest gratitude to my advisors, Professor Yu Hen Hu and Professor Robert Radwin, for their endless support, inspiration, and guidance throughout my Ph.D. study. Many thanks to other project collaborating faculty members: Professor John D. Lee for providing many chances to improve my presentation skills and critiques; Dr. Jack Lu for providing many large-sized essential data.

Thanks to my thesis committee members, Professor Bill Sethares, Professor John A. Gubner, and Professor Vikas Singh, for their suggestions, critiques, and help shaping the direction of this dissertation.

Special thanks to the members of the Occupational Ergonomics and Biomechanics Lab, Runyu L. Greene, David Azari, Oguz Akkas and Zishuai Zou for providing valuable help for carrying out experiments and making comments on my progress report. Also special thanks to my labmate Jian Wei Ke for numerous discussion on 3D reconstruction techniques. I'd also like to thank ZhengYang Lou for contributing to my pose estimation paper.

Great thanks to all of my sincere friends: Huilong Zhang, Hanwen Chen, Haoran Yu, Na Li, Mengguo Jing, Xuan Zhang, Yinghan Xu, Xue Yin, Yuzhang Zang, Hong Jiang, Xiaoting Yang, Pei Yang, and Yue Shao. Thanks for their spiritual support that accompanies me overcoming countless difficulties during my Ph.D. study.

Last but not least, I would like to express my sincere gratitude to my parents for their guidance, full support, and love, which has never left and will always be with me.

To my parents

Contents

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

In this research project, we develop efficient Video Object Tracking (VOT) algorithms and video behavioral analysis algorithms for non-invasive observing and analyzing human activities. Two specific applications are investigated: monitoring driver distraction and ergonomics of factory workers performing heavy lifting jobs.

We develop VOT algorithms to process a large scale SHRP-2 Naturalist Driving Study (NDS) dataset. The NDS dataset contains hundreds of thousands of hours of videos monitoring thousands of drivers driving on the road. The objective is to detect distractive driving events by processing billions of video frames. A key research effort is to develop a VOT algorithm that can significantly reduce the computing time without compromising the overall tracking performance. To this aim, we developed a novel temporal frame-subsampling (TFS) algorithm. Instead of processing every frame, the TFS algorithm performs video object tracking on a sub-sampled shorter video sequence obtained by sampling 1 frame out of N consecutive video frames from the original video sequence. The VOT tracking results (object's locations) of the N-1 frames between two consecutive sampled video frames will be obtained through interpolation of tracking results of the sampled frames. We show that for some NDS video sequences, N can be as large as 100 (100-fold speed-up) without sacrificing the VOT performance.

The second application concerns body posture monitoring of industrial workers performing heavy lifting jobs. We developed a motion-segmentation based bounding box tracking algorithm to track the movement of a worker during the lifting operation. We show that the variations of the size of the bounding box can be used to infer the body posture (bending angle) of the worker during the lifting instant. Specific lifting and dropping events are detected by exploiting a ghosting phenomenon when the position of a lifted object changes from stationary to motion. This work validates the feasibility of using light-weight video analytics to estimate body posture without resorting to cumbersome and expensive 3D tracking devices.

Along this direction, we also developed novel algorithms to estimate the body twisting (asymmetry) angle of the worker during a lifting event. The asymmetry angle is defined on the relative positions of the worker's wrists and ankles which are part of the skeletal joints of the worker's body. We incorporate state of art 2D and 3D skeletal joint estimation algorithm to aid the detection of 3D body joints locations. Then, we deduce the corresponding asymmetry angles. This algorithm is tested with a large dataset from a NIOSH laboratory. Very promising results have been observed.

The works accomplished in this research provide strong evidence that a non-invasive video-based

approach can provide accurate estimates of the states of human drivers during driving or workers during working. The algorithms developed in this work built on state-of-art computer vision algorithms. The main contributions of this work lie in the development of computation efficient formulation without losing performance.

Chapter 1 Introduction

1.1 Video-based Human Activities Analysis

Human activities analysis facilitates automated monitoring and understanding of human behaviors. By analyzing the human body movements, athletes and factory workers may improve skills, and avoid injuries. Tracking human movements and actions in a public space may help to detect security threats. By tracking the driver's head movement during driving, distractive driving conditions may be detected early to avoid traffic accidents. These are just some examples of the vast number of potential applications of human activity analysis.

With the rapid advancement of computer vision hardware, such as digital cameras, and depth sensors, our ability to acquire high-quality video at low cost improves significantly. We also witness great enhancement of computer vision algorithms for object detection, object recognition, object tracking, and image and video segmentation, among many other tasks. Despite these signs of progress, fully automated human activity analysis, however, is still not yet accomplished. The movements of the human of interests need to be tracked visually from a given video. Then, depending on applications, relative positions of different parts of the body (e.g. torso, limb, arms, hands, etc.) will need to be tracked. Sometimes, the contact of body parts with an object may be of interests for monitoring and the event of such contact (or release) may also need to be detected. Some of the latest computer vision algorithms may accomplish portions of these tasks off-line, and sometimes semi-automatically. Yet, these algorithms often involve extensive computation and yield results may not be immediately useful

for specific applications.

In the past, human activities monitoring is often performed manually by an operator watching a video and recording timestamps of different events of interests. Or sometimes intrusive, cumbersome sensors and wires need to be placed on the body of the tester. In the past few years, using advanced computer vision algorithms, it is possible to track one or more *region of interests* (ROIs) in a video and estimate the trajectory of one or more parts of a human body with moderate success. Built on these earlier results, the main objective of this research is to develop *efficient*, domain-specific computer vision algorithms for two specific applications. By efficient, we aim at algorithms that will take less time to compute for very long video sequences and can also support real-time human activity analysis for videos taken in a factory environment.

The first application area is driver distraction detection. The task is to monitor the driver's head position during driving and detect events that the driver's head moves away from the normal driving positions as those events may be hints for driver distraction. A unique challenge in this task is the volume and lengths of the video clips that need to be processed. In a recent study sponsored by US Federal Highway Administration (FHA), a Second Strategic Highway Research Program (SHRP-2) Naturalist Driving Study (NDS) collection of videos contains more than hundreds of thousands of hours of video taken over several years on more than 3400 drivers driving on the road. Processing these many video clips certainly is a daunting task and demands special attention to seek a computationally efficient computer vision algorithm.

A second application area investigated in this research is the monitoring of manual lifting tasks in a factory work environment. Overexertion in manual lifting ranks first among the leading causes of disabling workplace injuries. In 2017, this type of work-related injuries costs businesses $13.79 billion in direct costs and accounted for 23 percent of the overall national burden [54]. The purpose of lifting monitoring is to detect unsafe workflow and make corrections before injuries occur. National Institute of Occupational Safety and Health (NIOSH), a federal agency has provided a formula, the revised NIOSH lifting equation (RNLE), to estimate the upper bounds of health-safe lifting weights. The goal of our research is to estimate the key parameters of this formula accurately based on the video recording of the lifting operations in the factory. A requirement is to be able to develop light-weight computer vision algorithms so that they may be implemented on mobile devices like a cell phone while still providing real-time monitoring capability.

1.2 Motivation

Although a lot of progress has been achieved in video object tracking, VOT tasks for long-term (hour-long) video sequences have not been extensively investigated. Existing VOT benchmark OTB [8] or VOT challenges [9] - [13] focus on enhancing tracking performance for short videos that are no more than a couple of minutes in length. The computation time, while documented and compared, is not a major concern in these studies. Previously, a Long-Term Detection and Tracking (LTDT) [15] task was defined for video sequences at least 2 minutes long (at 25-30 fps), but ideally 10 minutes or longer. VOT Challenge 2018 [81] initiates the competition for long-term video tracking, but it is just emphasizing the

occlusion and the target leaving out of the image. The tracker performance is more focused on the effectiveness of re-detection. The computing time and the drifting issue still haven't got much attention. Thus, we launched our research to solve the long-term video tracking problem, with the goal of reducing computation load while keeping high accuracy.

In the researching area of human motion recognition, people have demonstrated a lot of great work by using wearable sensors, 2D cameras, and RGB-D cameras. However, they are suffering from the problem of intrusive equipment or a high-demand and complex computation. In the field of manual lifting tasks, despite the great number of overexertion cases in manual lifting tasks, there is no low-costing and convenient tool that can be prevalently used for predicting lifting risks. The problem of lifting risks hasn’t been intervened well. So we planned to propose an efficient tool, which just relies on normal cameras and low-demand computation to analyze the human lifting behavior and give out a risks prediction.

1.3 Objective and Contributions

For the project of driver behavior analysis, the objective was to investigate a tracking scheme that could work well on long-term practical video data. The goal was to reduce object tracking processing time and keep high and robust accuracy. The tracker would be based on the sequential Bayesian tracking framework and employ sub-sampling scheme.

The technical contributions include a detailed analysis of the temporal frame sub-subsampling

scheme for video object tracking and a set of empirical rules for the proper application of Temporal Frame Sub-sampling (TFS) to significantly reduce VOT processing time while enhancing VOT accuracy. And, a drift-resilience scheme has been proposed. Besides, a set of driver behavior analyzing tool has been implemented. This work has resulted in three conference papers and one journal paper:

- Xuan Wang, Yuhen Hu, Robert G. Radwin, John D. Lee, “Frame-Subsampled, Drift-Resilient Video Object Tracking”, *2018 International Conference on Acoustics, Speech, and Signal Processing* (IEEE ICASSP 2018) [16]

- Xuan Wang, Yuhen Hu, Robert G. Radwin, John D. Lee, “Frame-Subsampled, Drift-Resilient Long-Term Video Object Tracking”, *IEEE International Conference on Multimedia and Expo* (ICME) 2018 [17]

- Xuan Wang, Yuhen Hu, Robert G. Radwin, John D. Lee, “Head Tracking Using Video Analytics”, *IEEE Global Conference on Signal and Information Processing 2015* (Global SIP) [71]

- Xuan Wang, Yuhen Hu, Robert G. Radwin, John D. Lee, "Temporal Frame Sub-Sampling for Video Object Tracking", *Journal of Signal Processing Systems* (2019): 1-13.[18]

For the manual lifting risks monitoring project, the objective was to propose a robust, non-intrusive, straightforward approach to automatically extract spatial and temporal factors necessary for applying the RNLE using two video camera views of the subject.

The contributions include an efficient and practical approach to automatically extract spatial and temporal factors necessary for applying the RNLE. It contains two parts. The first part leverages motion information to detect lifting instants and the hand and ankle locations during lifting with one camera looking into the subject's sagittal plane. This proposed method can be used to evaluate a work task with fewer computations and sufficiently high accuracy when compared with 3D camera approaches. It has resulted in a recent patent application filed by the Wisconsin Alumni Research Foundation:

- Robert G. Radwin, Xuan Wang, Yu Hen Hu, and Nicholas Difranco, “Movement Monitoring System”, P170280US02 [94]

Also, a journal paper:

- Xuan Wang, Yuhen Hu, Ming-Lun Lu, Robert G. Radwin, " The Accuracy of a 2D Video-Based Lifting Monitor", *Ergonomics* just-accepted (2019): 1-33.[19]

The algorithm has been utilized in the work:

- Greene, R.L., Hu, Y.H., Difranco, N., Wang, X., Lu., M.L., Bao,S., Lin, J.H., and Radwin, R.G. (2018, August). "Predicting Sagittal Plane Lifting Postures from Image Bounding Box Dimensions." *20th Congress of International Ergonomics Association*. Florence, Italy. [112]

- Greene, R.L., Hu, Y. H., Difranco, N., Wang, X., Lu, M. L., Bao, S., Lin, J.-H .& Radwin, R. G. (2018, August). Automated Video Lifting Posture Classification Using Bounding Box Dimensions. In Congress of the International Ergonomics Association (pp. 550-552). Springer,

Cham. [113]

- Greene, R. L., Hu, Y. H., Difranco, N., Wang, X., Lu, M. L., Bao, S. Lin, J.-H. & Radwin, R. G. (2019). Predicting Sagittal Plane Lifting Postures From Image Bounding Box Dimensions.Human factors, 61(1), 64-77. [114]

Another part of this work is supported by a proposal to NIOSH Michigan Pilot Project Training Program, which has been accepted. It estimates the asymmetry angle for the revised the RNLE by combining structure from motion and CNN based 2D skeletal model estimator. This work results in a journal paper titled "Body asymmetry angle estimation based on computer vision approach", which we are currently working on.

1.4 Organization

The organization of this report is as follows:

In Chapter 2, the traditional video object tracking methods and video-based ergonomics methods are discussed. The concept of sequential Bayesian tracking and the tracking performance measurement is introduced.

In Chapter 3, we describe a frame subsampled tracking scheme to help save computation load for long-term video, but at the same time, an accuracy increase has achieved. The reasons and the principles of getting this accuracy improvement are discussed. Besides, we also propose a drift reliance scheme. The proposed drifting probability helps to supervise the tracking progress and save the tracking from

following a wrong template and wasting computation source. Also, tools for analyzing a driver's behavior are provided, including hand state detection and road state detection. Together with head tracking, this helps give a comprehensive driver behavior analysis.

In Chapter 4, we propose a robust, non-intrusive, straightforward approach to automatically extract spatial and temporal features of a subject performing lifting tasks. These factors are necessary for applying the NIOSH lifting equation. It only relies on a single video camera view of the subject's sagittal plane. This approach helps to monitor the overexertion risks during lifting tasks. This lifting monitor algorithm takes advantage of motion information to distinguish the moving subject from the static background. The experiment shows it achieves a highly accurate performance.

In Chapter 5, we extend the work in Chapter 4 and propose a body asymmetry angle estimation algorithm. We realized this by a fusion of the 2D skeletal joints estimation and the technology of structure from motion. Instead of depth cameras, 2 normal RGB cameras are used. The 2D skeletal joints estimations on each view of the camera are used as feature points for 3D reconstruction. The experiment shows promising results. This algorithm together with the previous work of estimating spatial and temporal factors using motion-based information, could be used to automatically calculate the RNLE. The experiments conducted show that this approach provides a high-accuracy estimation, making a hand-held automatic and convenient lifting risk predictor realizable.

In Chapter 6, we conclude that the proposed methods are contributing to the practical video-based human activity analysis. Possible improving approaches are discussed.

Chapter 2 Conventional Human Activities Analysis by Video

2.1 Traditional Video Object Tracking Methods

Video object tracking has become a hot topic with the development of camera and computer vision technologies. And it has achieved great progress for different types of trackers. In this preliminary report, the focus is put on the topic of tracking long-term videos and saving computation load.

Video object trackers are categorized into generative and discriminative types. The traditional generative category is represented by Kalman filter, particle filter, and mean-shift. Most of them are relatively weak in accuracy when compared with the state-of-the-art trackers. But recently, there is an outstanding state-of-art generative tracker, ASMS [93], which proposes background ratio weighting to exploit the neighborhood to help discriminate the target, and relies on forward-backward consistency and scale expansion and implosion regulation to do scale selection. It outperforms the contemporary trackers in both accuracy and speed. And it claims an average speed of 125 fps.

With the development of visual classification, the discriminative category, which is also called tracking by detection, applying machine learning with a good selection of features onto computer vision, becomes the main method for nowadays video object tracking.

The classical discriminative tracker, STRUCK [5], uses kernelized structured output support vector machines to select and learn from targets online. It is recognized as a representative of the

best-performance trackers of the contemporary trackers. With a high accuracy rate, its speed is claimed to be 20 fps.

The classical tracker for long-term tracking, TLD [3], explicitly decomposes the long-term tracking task into three sub-tasks: tracking, learning, and detection. The detector corrects the tracker if necessary. And the learner estimates the detector's errors and learns from them to update the detector. This self-correction scheme facilitates itself to be applicable to long-term video tracking. As a classical tracker, its tracking speed is claimed as 28 fps.

The nowadays most popular trend of tracking, correlation filter based trackers, have now been recognized as the fastest tracker and achieve outstanding accuracy. It has intrinsically escalated tracker speed from real-time to high speed by leveraging Fourier transformation in the calculation step. MOSSE [92], the originator of applying correlation filter into tracking, uses a new type of correlation filter, minimum output sum of squared error filter, and claims a high processing speed of 615 fps with robust performance. KCF[20] is based on the correlation filter, and it proposes an analytic model and diagonalizes the circulant matrix with Fourier transform to reduce both storage and computation. Its accuracy is enhanced compared with MOSSE while its speed is sacrificed to be 172 fps.

Another popular trend is applying the convolutional neural network into video object tracking. Due to the huge framework of the convolutional neural network, the computation load is always very heavy and thus the tracking speed is not fast. SiamFC [90] is outstanding with a relatively high speed of 80 fps. It equips a basic tracking algorithm with a novel fully-convolutional Siamese network trained end-to-end

on the ILSVRC15 [91] for object detection in video. It doesn't perform any model update and uses only the first frame to compute the convolutional embedding function.

The existing efforts focused on reducing the computational complexity of the algorithm. However, none of these existing efforts mention whether the workload (number of video frames) may be reduced to decrease the overall cost of video object tracking.

2.2 Conventional Video-based Ergonomics Approaches

In this section, we focus especially on the manual lifting risks monitoring task. There are several tools provided to assess the lifting tasks. The revised NIOSH lifting equation (RNLE) [62] is the most widely used tool to assess the risk of low back pain associated with lifting and lowering tasks in the workplace [55]. The recommended weight limit (RWL), calculated by the RNLE, is defined for a specific set of task conditions as the weight of the load that nearly all healthy workers could perform over a substantial period of time without an increased risk of developing lifting-related low back pain. The RWL is implemented in the International Organization for Standardization (ISO) standard 11228 Part 1 Manual Lifting and Carrying, and widely used by practitioners for prevention of lifting-related low back pain. The RWL is calculated as:

$$\text{RWL} = LC \times HM \times VM \times DM \times AM \times FM \times CM \qquad (2.1)$$

where LC is a load constant equal to 23kg, HM, VM, and DM are a function of the distance between the location of the hands and feet at the origin and destination of the lift, AM is a function of

the angle of torso rotation in relation to the feet, FM is a function of the frequency of lifting, and CM is a function of the quality of the hand-to-object coupling.

Today's manual materials handling jobs have evolved from mono tasks decades ago to multiple and varying tasks, specifically in the manufacturing and transportation sectors, such as warehousing, distribution centers, package delivery trucks, baggage handling, lean or just-in-time manufacturing, kitting, palletizing and shipping. For prevention of work-related low back pain, it is, therefore, becoming increasingly important to frequently or continuously monitor workers' exposure to physical demands for varying job tasks.

Manually measuring the dimensions needed to calculate the RNLE is challenging, particularly in situations where lifting occurs in numerous locations involving varying body postures throughout the workday [48]. Direct measurement and instrument-based methods have been tested that employ sensors or markers attached to the subject for measuring certain variables ([51], [52] and [53]). However, these methods suffer from the invasiveness of the measurement, the space needed for equipment, and high cost [56].

With the advancement of video technologies, cameras have become a prevalent, low-cost, highly efficient, and non-intrusive option for monitoring ergonomic aspects of job tasks. Both 2D (RGB) and depth (RGB-D) cameras have attracted researchers' attention for developing direct-reading technologies for ergonomic risk assessments. In recent years, the development of depth cameras, e.g. Kinect (Microsoft, Redmond, WA), has stimulated a new trend in measuring human body dimensions. These

approaches rely on matching the recognized human body parts into a pre-trained skeletal model, which consists of the position of joints and body linkages [49]. Previous researchers used conventional 2D cameras for creating a skeletal model from a large image dataset ([45], [50], [58], [59], [60]).

Although 2D cameras are more prevalent than depth cameras, 2D cameras propose a more challenging partial body recognition problem. With one more dimension of information, RGB-D camera-based methods have been shown to provide more accuracy than RGB cameras ([56], [61], [47], [57]). Two studies ([56], [61]) used the Kinect to measure RNLE parameters. However, these Kinect based approaches were limited when there was occlusion; namely, the Kinect did not capture a frontal view of the subject.

2.3 Sequential Bayesian Tracking

Video object tracking is often presented in a sequential Bayesian estimation framework: Denote $\mathbf{x}_k$ to be the state (positions and speed) of the object being tracked in the k^{th} frame, $\mathbf{z}_{1:k}$ be the observations (video frames) from the first to the k^{th} frame. Our goal is to estimate the posterior probability $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ sequentially using estimated state probability distribution at the previous frame $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$, the state transition probability $p(\mathbf{x}_k|\mathbf{x}_{k-1})$, and the likelihood function at the k^{th} frame $p(\mathbf{z}_k|\mathbf{x}_k)$. It is divided into two steps:

$$\text{Predict: } p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})dx_{k-1} \tag{2.2}$$

$$\text{Update: } p(x_k|z_{1:k}) = \frac{p(z_k|x_k)p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})} \propto p(z_k|x_k)p(x_k|z_{1:k-1}) \tag{2.3}$$

$p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ is the predicted object state at the k^{th} frame. In many existing VOT algorithms, it is approximated by a rectangular *search region* centered at estimated object state in the $(k\text{-}1)^{th}$ frame. The maximum a posterior (MAP) estimation of the object state at the k^{th} frame then will be found as

$$\hat{\mathbf{x}}_{k,MAP} = \max_{\mathbf{x}_k} p\left(\mathbf{x}_k \left| \mathbf{z}_{1:k} \right.\right) \tag{2.4}$$

2.3.1 Search Region

In VOT literature, the prior distribution $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ is often characterized by a search region within a video frame. A search region is a sub-region of the entire video frame within which the object may be found. The size of the search region dictates the computation requirement and impacts on the tracking accuracy. Theoretically, the search region may be defined as:

$$R_s = \left\{ X_k \left| p\left(X_k \left| Z_{1:k-1} \right.\right) \geq \varepsilon \right.\right\} \tag{2.5}$$

From Eq. (2.2), given $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$ and $p(\mathbf{x}_k|\mathbf{x}_{k-1})$, $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ can be analytically evaluated. In practice, few VOT tracking algorithms estimate $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$ explicitly. Instead, it is often implicitly assumed

$$p\left(X_{k-1} \left| Z_{1:k-1} \right.\right) \sim \delta\left(X_{k-1} - \hat{X}_{k-1} \right) \tag{2.6}$$

Substituting into Eq. (2.2), we have $p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1})$. Therefore, Eq. (2.5) becomes

$$R_s = \left\{ X_k \left| p\left(X_k \left| \hat{X}_{k-1} \right.\right) \geq \varepsilon \right.\right\} \tag{2.7}$$

Different models of the state transition probability $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ lead to different estimations of the search region.

2.3.2 Likelihood

Thus, the likelihood $p(\mathbf{z}_k|\mathbf{x}_k)$ can be interpreted as the similarity of the candidate template at $\mathbf{x}_k$ from the current frame $\mathbf{z}_k$ to that of the provided template. We shall denote $T(\mathbf{z}_k|\mathbf{x}_k)$ as the template (of the similar size as the provided template) and T_0 as the one-shot template provided at the beginning. In other words,

$$p\left(Z_k \middle| X_k\right) \propto g\left(T\left(Z_k \middle| X_k\right), T_0\right)$$

The template T_0 and candidate matching template $T(\mathbf{z}_k|\mathbf{x}_k)$ may be provided as RGB pixel values over the template region which can be regularly shaped, such as a rectangle, or irregularly shaped if the image of the object is segmented to be separated from the background.

2.4 Performance Measurement

2.4.1 Accuracy Measurement

The performance of VOT may be evaluated using several different criteria such as precision plot and success plot [8]. In this work, we use the area overlap ratio (AOR) R defined in (2.8) to determine if the tracking is still on track or already drifted to gauge the tracking performance:

$$R = \frac{\left|S_{GT} \cap S_{est}\right|}{\left|S_{GT} \cup S_{est}\right|} \geq t \tag{2.8}$$

where $\cap$ and $\cup$ represent the intersection and union of the area in the ground truth bounding box S_{GT} and the area in the estimated bounding box S_{est}. According to [8] and [31], in this paper, we set

threshold $t = 0.5$. If $R \geq 0.5$, it is deemed a correct tracking (success). Otherwise, the tracking fails for that frame. We define the accuracy rate as the percentage of frames in a video sequence that are deemed as a success.

Besides AOR, we also use the distance from ground truth center to estimation center to measure how much the drift is. Though it doesn't count in the target size, it is the most direct way to see how much the estimation is biased from the ground truth. So we use center distance in the analysis, but use the more comprehensive AOR in the experiment.

2.4.2 Speed Measurement

VOT processing speed in the unit of frames per second (fps) has been one of the performance metrics in recent video object tracking challenges [8]. In VOT 2014 Challenge [10], a metric called the equivalent filter operation (EFO) is used to compare the processing time of different VOT algorithms. However, with frame sub-sampling, we opt to use the overall VOT processing time as a more direct metric.

Chapter 3 Video-based Driver Behavior Analysis

As with the prevalence of vehicles, traffic safety has become a significant issue in people's daily life. According to U.S. Department of Transportation, National Highway Traffic Safety Administration, in 2017, there were an estimated 6,452,000 police-reported traffic crashes, in which 37,133 people were killed and an estimated 2,746,000 people were injured. One person was killed every 14 minutes and an estimated 5 people were injured every minute in motor vehicle crashes in 2017. A lot of effort has been carried out on this issue. [63] [64] [65] [66] analyzes the effect of the road environment to the driving safety issue, [67] studied from the aspect of vehicle motion, [68] analyzes the driver's behavior. In this work, the role of the driver's behavior in the traffic safety is studied.

3.1 Temporal Frame Subsampled (TFS) Tracking

For the goal of saving the computation load of long-term video tracking, we propose Temporal Frame Subsampled (TFS) tracking.

3.1.1 Frame Sub-sampling and Error Estimation

Frame sub-sampling exploits temporal correlation of object motion in a video sequence. By frame sub-sampling at a ratio of N, one frame out of every consecutive N frames will be extracted to form a length-reduced video sequence (at a ratio of N: 1). A VOT algorithm then will be applied to track the object using this length-reduced video sequence. This yields an estimated object position at each of the sub-sampled frames. The positions of the object in the N–1 frames between a pair of adjacent

sub-sampled frames then will be estimated based on the position estimates in these sub-sampled frames using linear interpolation. It is reasonable to assume the computing time for position interpolation will be orders of magnitude less than that for VOT per frame. Therefore, using frame sub-sampling, the total computation time may be reduced N times, representing orders of magnitude speed-up.

In earlier preliminary works [16] and [17], it is observed that when frame sub-sampling is applied to certain video sequences using certain VOT trackers, not only the computing time is reduced by N fold, the tracking accuracy is also significantly improved. In this work, our focus will be to provide an in-depth analysis of this accuracy enhancement phenomenon of the frame sub-sampling scheme. Two fundamental questions will be investigated: (a) What causes the enhancement of tracking accuracy when the frame sub-sampling scheme is applied? (b) How to determine if frame sub-sampling will yield performance enhancement in addition to computing time reduction for a given video sequence and a specific VOT algorithm?

To facilitate an empirical study of the above questions, we will use 60 training video clips from the 2016 video object tracking challenge [12] and corresponding annotation of object motion tracks (ground truth). Below, we will investigate interpolation error and its relations to the object trajectory.

To analyze the tracking error, let the (true) position of the object being tracked at the n^{th} frame be denoted as $\mathbf{x}_n$ and the estimated position as $\hat{\mathbf{x}}_n$. With a sub-sampling ratio of N, $\hat{\mathbf{x}}_0$ and $\hat{\mathbf{x}}_N$ will be estimated using a VOT algorithm from the sub-sampled video sequence and $\{\hat{\mathbf{x}}_n;\ 1 \le n \le N-1\}$ will be estimated using linear interpolation:

$$\hat{\mathbf{x}}_n = (1 - n/N)\hat{\mathbf{x}}_0 + (n/N)\hat{\mathbf{x}}_N \quad 1 \le n \le N-1. \tag{3.1}$$

Denote $\mathbf{e}_n = \mathbf{x}_n - \hat{\mathbf{x}}_n$ to be the tracking error at the n^{th} frame.

$$\begin{aligned} \mathbf{e}_n &= \mathbf{x}_n - \left[(1 - n/N)\hat{\mathbf{x}}_0 + (n/N)\hat{\mathbf{x}}_N\right] \\ &= \left(\mathbf{x}_n - \left[\left(1 - \frac{n}{N}\right)\mathbf{x}_0 + \frac{n}{N}\mathbf{x}_N\right]\right) + \left[\left(1 - \frac{n}{N}\right)\mathbf{e}_0 + \frac{n}{N}\mathbf{e}_N\right] \\ &= \mathbf{e}_{n,\text{int}} + \mathbf{e}_{n,est} \end{aligned} \tag{3.2}$$

where $\mathbf{e}_0 = \mathbf{x}_0 - \hat{\mathbf{x}}_0$ and $\mathbf{e}_N = \mathbf{x}_N - \hat{\mathbf{x}}_N$.

In (3.2), the interpolation error $\mathbf{e}_{n,int}$ depends on the object motion pattern, and $\mathbf{e}_{n,est}$ depends on the tracking accuracy of the VOT algorithm.

3.1.1.1 Interpolation Error

Let $\{k;\ 0 \le k \le K-1\}$ be the indices of the sub-sampled frames. At a sub-sampling factor N, the original indices of these frames would be $\{n;\ n = n_0 + kN,\ 0 \le k \le K-1\}$. Here n_0, $0 \le n_0 < N$ is the index of the first sub-sampled frame. Let x_n denote the x-dimension of $\mathbf{x}_n$. Consider a sub-sampling sequence

$$s[n] = \sum_{k=0}^{K-1} \delta[n - n_0 - kN] = \frac{1}{N}\sum_{\ell=0}^{N-1} e^{j2\pi(n-n_0)\ell/N} \quad 0 \le n < KN \tag{3.3}$$

where $\delta[n] = 1$ if $n = 0$ and 0, otherwise. Define

$$\tilde{x}[n] = x[n] \cdot s[n] = \begin{cases} x_n & n = n_0 + kN,\ 0 \le k \le K-1, \\ 0 & otherwise. \end{cases} \tag{3.4}$$

The trajectory of the frame sub-sampled video sequence then may be expressed as $\{\tilde{x}[n_0 + kN];\ 0 \le k \le K-1\}$. Let $\tilde{X}[m]$ be the discrete Fourier transform (DFT) of $\tilde{x}[n]$. Then for $0 \le m \le$

M–1, $M = K{\cdot}N$,

$$\tilde{X}[m] = \sum_{n=0}^{M-1} \tilde{x}[n] e^{-j2\pi nm/M} = \sum_{n=0}^{M-1} \left(x[n] \cdot s[n] \right) e^{-j2\pi nm/M}$$

$$= \sum_{n=0}^{M-1} x[n] \cdot \left(\frac{1}{N} \sum_{l=0}^{N-1} e^{j2\pi l (n-n_0)/N} \right) e^{-j2\pi nm/M}$$

$$= \frac{1}{N} \sum_{l=0}^{N-1} e^{-j2\pi l n_0/N} \sum_{n=0}^{M-1} x[n] \cdot e^{-j2\pi n(m-lK)/M}$$

$$= \frac{1}{N} \sum_{l=0}^{N-1} e^{-j2\pi l n_0/N} X[m - lK] \qquad (3.5)$$

In other words, $\tilde{X}[m]$ is a weighted sum of $X[m-\ell K]$, $0 \le \ell < N$. The object trajectory for frames between successive sub-sampled frames then will be estimated using linear interpolation:

$$x_r[n] = \left(1 - \frac{n}{N} + k \right) \tilde{x}[kN] + \left(\frac{n}{N} - k \right) \tilde{x}[(k+1)N] \qquad (3.6)$$

where $k = \lfloor n/N \rfloor$ gives the integer portion of n/N. Define a triangular linear interpolation kernel

$$T[n] = \begin{cases} \frac{1}{N}\left(1 - \frac{|n|}{N} \right) & -N \le n \le N, \\ 0 & \textit{elsewhere.} \end{cases}$$

It is easily verified that

$$x_r[n] = \tilde{x}[n] ** T[n]$$

where "**" is the convolution operator. Thus, the KN-point DFT of $x_r[n]$ can be written as:

$$X_r[m] = \tilde{X}[m] \cdot T[m] \qquad 0 \le m \le KN-1,$$

where

$$T[m] = \begin{cases} 1 & m = kN \\ \frac{1}{N^2} \frac{\sin^2(\pi m)}{\sin^2(\pi m/N)} & m \ne kN \end{cases} \qquad (3.7)$$

Recall that interpolation error is

$$e_{n,\text{int}} = x_n - x_r[n].$$

Thus, the DFT of the interpolation error can be expressed analytically as (assuming $n_0 = 0$)

$$E_{\text{int}}[m] = X[m] - T[m] \cdot \frac{1}{N} \sum_{l=0}^{N-1} e^{-j2\pi l n_0 / N} X[m - lK] \tag{3.8}$$

Since $T[m]$ amounts to a low pass filter. To reduce $|E_{int}[m]|$, it would be desirable that the values of $|X[m]|$ diminish quickly as $m > K = M/N$ to reduce the aliasing effects in the pass-band.

The root-mean-square (RMS) interpolation error $\|\bar{e}_{\text{int}}\|$ (in units of pixels) can be estimated using (3.8) because according to the Parseval theorem [32]

$$\sum_{n=0}^{M-1} \left\| e_{n,\text{int}}[n] \right\|^2 = \frac{1}{M} \sum_{m=0}^{M-1} \left\| E_{\text{int}}[m] \right\|^2 = M \cdot \left\| \bar{e}_{\text{int}} \right\|^2 .$$

We remark that while the above analysis is for the x-dimension of the trajectory, these procedures should be applicable to the y-dimension trajectory as well. Then, the overall RMS interpolation error will be less than the sum of the RMS errors of both dimensions. That is,

$$\left\| \bar{e}_{\text{int}} \right\|^2 \le \left\| \bar{e}_{\text{int},x} \right\|^2 + \left\| \bar{e}_{\text{int},y} \right\|^2$$

For every long video sequence, $X[m]$ cannot be evaluated. Instead, a short segment training sequence taken from the long sequence can be used to estimate the shape of $X[m]$ and an estimate of $E_{int}[m]$.

Example 1. Using clips #22 and #23 of the VOT 2016 Challenge video set, the corresponding $\|\bar{e}_{\text{int}}\|$

values versus N are plotted in Figure 1. These results are identical using either (3.2) or (3.8). For clip #22, the interpolation error increases very slowly with respect to N. For clip #23, the interpolation errors increase rapidly and exhibit significant variations. This may be explained using the frequency spectra of the x, y trajectories of these two clips. If we use -20dB as a threshold to determine the bandwidth of these signals, it is apparent the x and y trajectories of clip #22 have a narrow bandwidth ≤ 0.1 Hz (Figure 2). On the other hand, the x and y trajectories of clip #23 have a bandwidth of almost 1 Hz. According to (3.8), clip #23 will suffer lots of aliasing noise and hence significant interpolation error if TFS is applied.

3.1.1.2 Approximation Error

In (3.2), note that

$$\begin{aligned}\|\mathbf{e}_{n,est}\|^2 &= \left\|\left(1-\frac{n}{N}\right)\mathbf{e}_0+\frac{n}{N}\mathbf{e}_N\right\|^2 \\ &= \left(1-\frac{n}{N}\right)^2\|\mathbf{e}_0\|^2+\left(\frac{n}{N}\right)^2\|\mathbf{e}_N\|^2+2\left(1-\frac{n}{N}\right)\left(\frac{n}{N}\right)\mathbf{e}_0^T\mathbf{e}_N\end{aligned}$$

Summing over n from 0 to N–1, one has

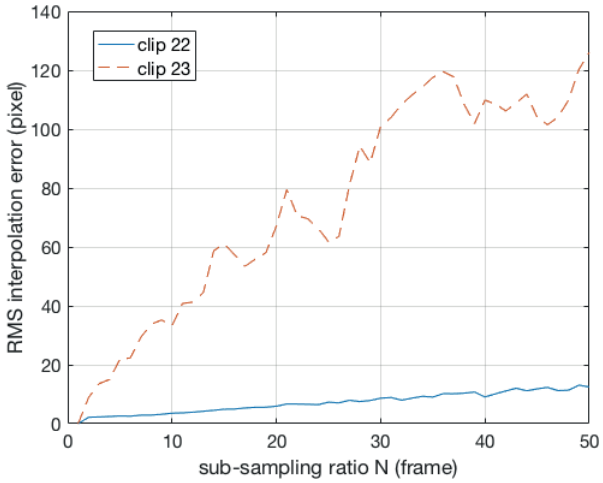


Figure 1 RMS interpolation error (pixel) versus N of clips #22, #23.

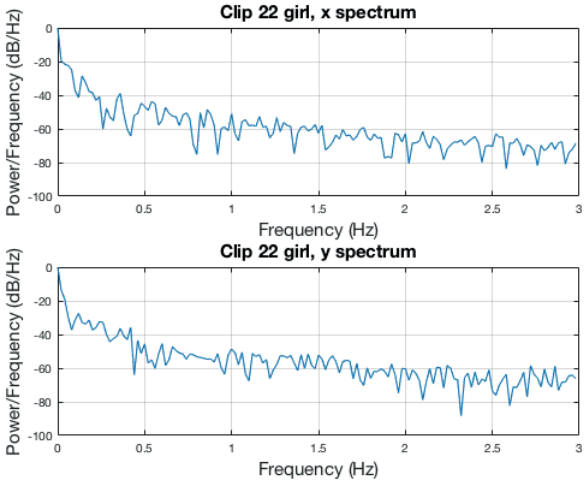


Figure 2 The frequency spectrum of the trajectory of clip #22. Note the steep drop beyond roughly 0.1 Hz. To focus on the low-frequency portion, only 1/6 of the horizontal frequency axis is shown.

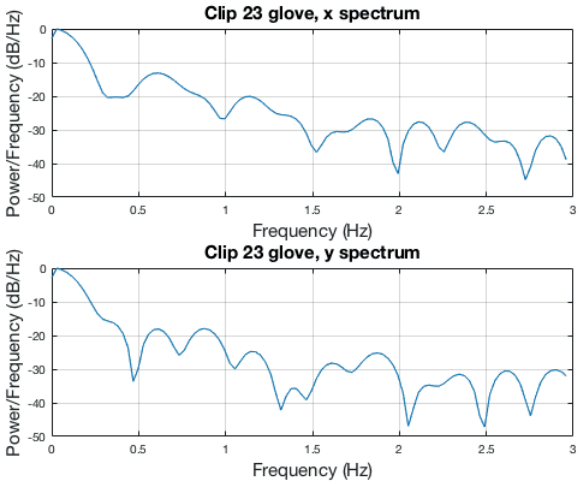


Figure 3 The frequency spectrum of ground truth trajectories of clip #23. Note the large sidebands up to almost 1 Hz. Zeros are padded to make the overall length of these sequences 1024.

$$\begin{aligned}\frac{1}{N}\sum_{n=0}^{N-1}\left\|\mathbf{e}_{n,est}\right\|^2 &= \left\|\mathbf{e}_0\right\|^2\cdot\frac{1}{N}\cdot\sum_{n=0}^{N-1}\left(1-\frac{n}{N}\right)^2+\left\|\mathbf{e}_N\right\|^2\cdot\frac{1}{N}\cdot\sum_{n=0}^{N-1}\left(\frac{n}{N}\right)^2\\ &\quad+2\mathbf{e}_0^T\mathbf{e}_N\cdot\frac{1}{N}\cdot\sum_{n=0}^{N-1}\left(1-\frac{n}{N}\right)\left(\frac{n}{N}\right)\\ &= \left\|\mathbf{e}_0\right\|^2\cdot\frac{(N+1)(2N+1)}{6N^2}+\left\|\mathbf{e}_N\right\|^2\cdot\frac{(N-1)(2N-1)}{6N^2}\\ &\quad+\mathbf{e}_0^T\mathbf{e}_N\cdot\frac{(N-1)(N+1)}{3N^2}\end{aligned} \tag{3.9}$$

Equation (3.9) gives the averaged estimation error (of center distance) per frame in one segment (between two sub-sampled frames) when sub-sampling and linear interpolation is used. The overall averaged estimation error of center distance can be expressed as:

$$\begin{aligned}\frac{1}{KN}\sum_{n=0}^{KN-1}\left\|\mathbf{e}_{n,est}\right\|^2 &= \frac{(N+1)(2N+1)}{6KN^2}\cdot\sum_{k=0}^{K-1}\left\|\mathbf{e}_{kN}\right\|^2\\ &\quad+\frac{(N-1)(2N-1)}{6KN^2}\cdot\sum_{k=1}^{K}\left\|\mathbf{e}_{kN}\right\|^2+\frac{(N-1)(N+1)}{3KN^2}\cdot\sum_{k=0}^{K-1}\mathbf{e}_{kN}^T\mathbf{e}_{(k+1)N}\end{aligned}$$

$$= \frac{(N+1)(2N+1)}{6KN^2} \cdot \left\|\mathbf{e}_0\right\|^2 + \frac{(N-1)(2N-1)}{6KN^2} \cdot \left\|\mathbf{e}_{KN}\right\|^2 + \frac{2N^2+1}{3KN^2} \cdot \sum_{k=1}^{K-1} \left\|\mathbf{e}_{kN}\right\|^2 + \frac{N^2-1}{3KN^2} \cdot \sum_{k=0}^{K-1} \mathbf{e}_{kN}^T \mathbf{e}_{(k+1)N} \tag{3.10}$$

With $N = 1$ (no TFS), the above expression becomes:

$$\frac{1}{K} \sum_{n=0}^{K-1} \left\|\mathbf{e}_{n,est}\right\|^2 = \frac{1}{K} \cdot \left\|\mathbf{e}_0\right\|^2 + \frac{1}{K} \cdot \sum_{k=1}^{K-1} \left\|\mathbf{e}_k\right\|^2 = \frac{1}{K} \cdot \sum_{k=0}^{K-1} \left\|\mathbf{e}_k\right\|^2 \tag{3.11}$$

As N increases, (3.10) can be approximately expressed as:

$$\begin{aligned} \left\|\bar{\mathbf{e}}_{est}\right\|^2 &= \frac{1}{KN} \sum_{n=0}^{KN-1} \left\|\mathbf{e}_{n,est}\right\|^2 \\ &\approx \frac{1}{3K} \cdot \left\{ \left\|\mathbf{e}_0\right\|^2 + \left\|\mathbf{e}_{KN}\right\|^2 + 2 \cdot \sum_{k=1}^{K-1} \left\|\mathbf{e}_{kN}\right\|^2 + \sum_{k=0}^{K-1} \mathbf{e}_{kN}^T \mathbf{e}_{(k+1)N} \right\} \\ &\le \frac{1}{K} \sum_{k=0}^{K-1} \left\|\mathbf{e}_{kN}\right\|^2 \triangleq \left\|\bar{\mathbf{e}}_{ss,N}\right\|^2 \qquad ss: \text{ sub-sampled} \end{aligned} \tag{3.12}$$

The significance of (3.12) is that the RMS estimation error $\|\bar{\mathbf{e}}_{est}\|$ of the interpolated sequence $x_r[n]$ does not increase the averaged distance estimation error per frame over that of the sub-sampled sequence. However, the estimation error per frame of the sub-sampled sequence, namely, $\|\bar{\mathbf{e}}_{ss,N}\|$ may be larger for some values of N, and smaller for others.

3.1.1.3 Total Error Due to Sub-sampling

From (3.2), $\mathbf{e}_n = \mathbf{e}_{n,\text{int}} + \mathbf{e}_{n,est}$, hence

$$\left\|\mathbf{e}_n\right\|^2 = \left\|\mathbf{e}_{n,\text{int}} + \mathbf{e}_{n,est}\right\|^2 \le 2 \cdot \left(\left\|\mathbf{e}_{n,\text{int}}\right\|^2 + \left\|\mathbf{e}_{n,est}\right\|^2 \right).$$

Thus, the averaged tracking error per frame can be bounded as

$$\left\|\bar{\mathbf{e}}\right\|^2 \le 2 \cdot \left(\left\|\bar{\mathbf{e}}_{\text{int}}\right\|^2 + \left\|\bar{\mathbf{e}}_{est}\right\|^2 \right) \tag{3.13}$$

Example 2. We apply a classical correlation filter based VOT tracker *KCF* [20] to clips #22, and #23 respectively for *N* varying from 1 to 50. KCF uses known and estimated target images to train the filter and predicts the target location in the coming frame by locating the maximum value in the response map. In Figure 4and Figure 5, the overall tracking errors combining interpolation error and estimation error are plotted in terms of (a) error distance (3.13), and (b) AOR (section II.B).

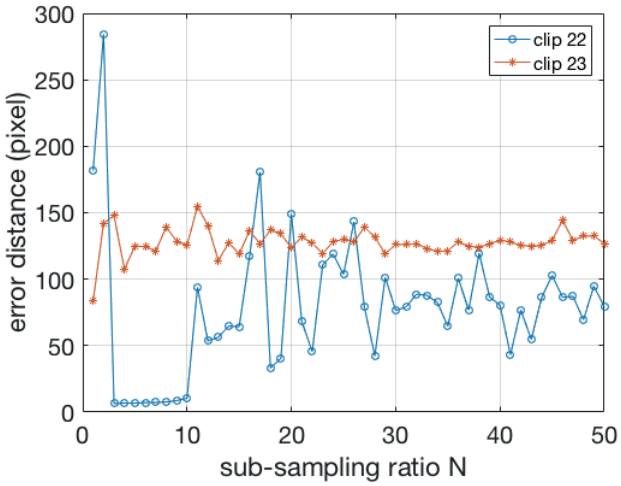


Figure 4 Error distance

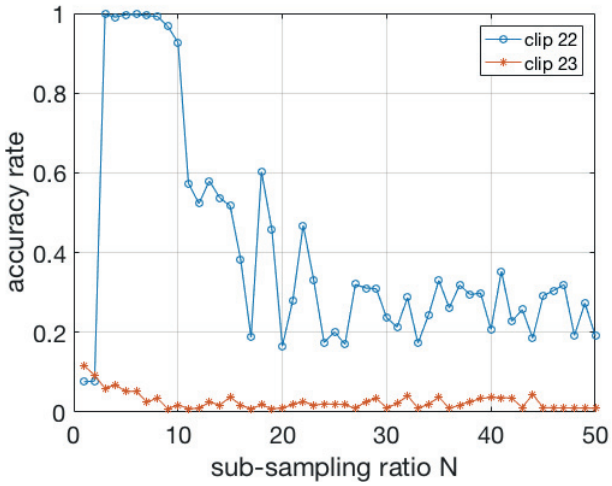


Figure 5 Accuracy rate by AOR

For video clip #22, using (3.12), we have $\|\bar{\mathbf{e}}_{ss,1}\| = 181.76$ for $N = 1$, and $\|\bar{\mathbf{e}}_{ss,8}\| = 7.60$ for $N = 8$. Also, the interpolation error $\|\bar{\mathbf{e}}_{\text{int}}\| = 0$ for $N = 1$ and $\|\bar{\mathbf{e}}_{\text{int},8}\| = 2.32$ for $N = 8$. Hence, as shown in Figure 4,

$$\|\bar{\mathbf{e}}_{\text{int},8}\| + \|\bar{\mathbf{e}}_{ss,8}\| = 9.92 < \|\bar{\mathbf{e}}_{ss,1}\| = 181.76.$$

On the other hand, for video clip #23, we have $\|\bar{\mathbf{e}}_{ss,1}\| = 83.76$, but $\|\bar{\mathbf{e}}_{ss,8}\| = 140.26$ and $\|\bar{\mathbf{e}}_{\text{int},8}\| = 24.79$. Hence,

$$\|\bar{\mathbf{e}}_{\text{int},8}\| + \|\bar{\mathbf{e}}_{ss,8}\| = 165.05 > \|\bar{\mathbf{e}}_{ss,1}\| = 83.76.$$

From Figure 5, we see two distinct tracking behaviors under different sub-sampling ratio N. The AOR performance of clip #22 improves significantly from < 0.1 for $N = 1$ and 2 to almost 1 (every frame has at least 50% area overlap) for $3 \le N \le 8$. This performance enhancement is unexpected. A detailed cause will be investigated in the next sub-section. On the other hand, for clip #23, which suffers significant interpolation error (Figure 1), the AOR performance decreases as the sub-sampling ratio N increase.

3.1.1.4 Performance Enhancement due to TFS

In Figure 4 and Figure 5, it has been observed that for $3 \le N \le 8$, the tracking performance of KCF on clip #22 has been dramatically improved over the cases of $N = 1$ or 2. To explore the root cause, in Figure 6, the error distances over the entire video clip #22 using KCF are plotted for all three cases $N = 1$, 2, and 3. Clearly, between the third and the fourth seconds, KCF lost track of the object for $N = 1$ and 2, but seems to remain on track (after a very brief deviation) for $N = 3$.

A closer look at the tracked object over these frames is in Figure 7. It reveals that the object, a girl wearing a yellow shirt on a skateboard, is covered by a white shirt male walking a bicycle with a pink basket starting from frame #105. At frame #109, the girl's image is completely occluded. The image of the girl reappears starting at frame #118. For the case of $N = 2$, at frame #111, the KCF tracker starts tracking the image of the person's white shirt and the pink basket of his bicycle instead of the yellow shirt girl. As such, the error distance suddenly increases starting from frame #111.

For the sub-sampled sequence of $N = 3$, the KCF tracker observes the beginning of occlusion at frame #106. After a brief deviation at frames #109, 112, and 115, it returns to the correct target at frame #118.

Another dramatic event in Figure 4 and Figure 5 is that the tracking accuracy drops abruptly between $N = 10$ and 11 for clip #22. In Figure 8 and Figure 9, we plot the tracking error distance against time for $N = 10$, and 11; as well as the template sequence between frame #870 and #1002. At the upper row of Figure 9, $N = 10$. The tracker loses track of the yellow-shirt girls in four frames #911, 921, 931, and 941. It recaptures the corrected target at frame #951. At the lower row ($N = 11$), the tracker latched on the wrong object (the man dressed in white) and never recovered from this mistake ever since.

Based on observations from Figure 4 to Figure 9, we realize the abrupt performance changes are due to the KCF tracker's decision to update the template of the object when the tracking object is temporarily affected by other objects. As explained in section 8.1 in [20], a new model is trained at the newly detected target position, and the model parameters will be linearly interpolated from those of the

previous frame. Hence the duration of the challenging events would affect whether the model parameters will be updated for a different (and in these cases, erroneous) object. Different frame sub-sampling ratios will affect the lengths of these challenging events. Whenever the KCF tracker mistakenly updates to an erroneous model, catastrophic performance degradations are observed. Unfortunately, without a training video available for these videos, it is rather difficult to determine what is the best value of sub-sampling ratio that would enhance the VOT performance most.

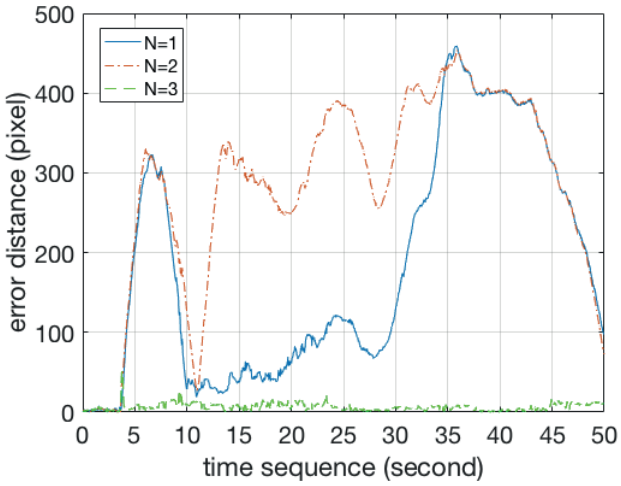


Figure 6 Tracking error distance per frame of clip #22 to analyze accuracy increase

Figure 7 Tracking result content for N=2 (first row) and N=3 (second row)

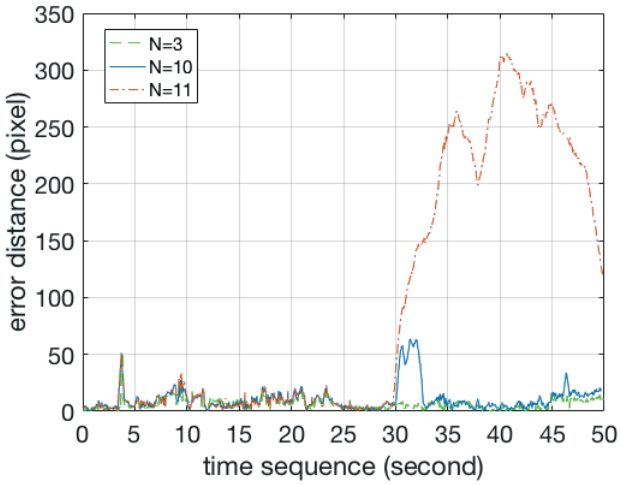


Figure 8 Tracking error distance per frame of clip #22 to analyze accuracy drop

Figure 9 Tracking result content for N=10 (first row) and N=11 (second row)

3.1.2 Applications, Discussions, and Extensions

3.1.2.1 Applications to VOT 2016 Challenge Video Dataset

We apply frame sub-sampling to the 60 video clips of VOT 2016 Challenge [12]. For each video clip of length M, we compute $K = \lfloor M/N \rfloor$ and take the first $KN+1$ frames for the sub-sampling experiment. The value of N will range from 1 to 50. Instead of computing the accuracy rate as in Figure 5, we compute the difference between the accuracy rate of each value of $N > 1$ versus the performance

when of $N = 1$.

In Figure 10, we give a 2D scatter plot where each point represents a value of N, $2 \leq N \leq 50$ using the KCF tracker. The horizontal axis represents the median value of performance difference (AOR(N) – AOR(1)). The vertical axis ranges from 0 to 1 representing the percentage of the 60 clips for that value of N that yields positive performance improvement when sub-sampling is used. From the figure, the horizontal axis ranges from -0.3 to 0, and the vertical axis ranges from 0.1 to 0.4. This implies the sub-sampling does not yield overall better performance than the original video ($N = 1$). This is expected because these are "challenged" videos often with rapid object motions. The data point at the upper right corner corresponds to $N = 4$. With this sub-sampling ratio, there are 21 clips exhibiting better performance. The lowest point corresponds to $N = 43$ and there are 9 videos (out of 60) exhibiting better performance. While the performance improvement due to sub-sampling alone is not universal, this experiment indicates that even for short and challenging videos like the visual object tracking challenge videos, sub-sampling is still an option that may worth exploring.

3.1.2.2 Applications to Long Video Sequences

Frame sub-sampling would be most useful when applied to very long video sequences to reduce computing time. With surveillance video recordings ranging from hours to days, it is affordable to manually annotate a few hundred frames as a training segment. Using this training video segment, one would want to decide whether to apply the temporal frame sub-sampling to reduce computation time and perhaps also realize performance enhancement. In this paper, we shall use three long video clips from

the SHRP 2 NDS collections to investigate these issues empirically.

We manually annotated the first hour (54000 frames at 15 frames/second) of two SHRP 2 videos HDEM24, and HDEM25, as well as a half-hour long (27000 frames) SHRP 2 video clip CHPV72. These videos recorded the same driving task on three different occasions. A typical frame of these video clips is shown in Figure 11. The objective of VOT is to track the driver's head position in these videos.

The hardware platform is a MacBook Air laptop computer with a 1.3 GHz Intel Core i5 processor. The main memory size is 4 GB with 1600 MHz. The trackers are implemented in C++ and compiled with standard C++11. The program runs under OS X EI Capitan operating system.

We use the first 2700 frames (180 seconds) of video clip HDEM24 as the training video.

First, we investigate the frequency spectra of the *x*, *y* trajectories of the object movement. In Figure 12, the frequency spectrum is plotted. The frequency spectrum dropped below -20dB within 0.1 Hz. Based on the discussion in section III.A, large frame sub-sampling ratio may be applied.

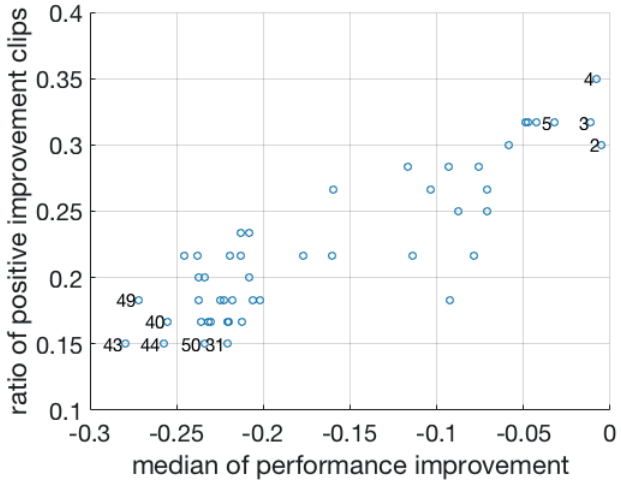


Figure 10 Performance improvement of each *N* with *N* marked for some extreme performance points

Figure 11 A demo video frame cut

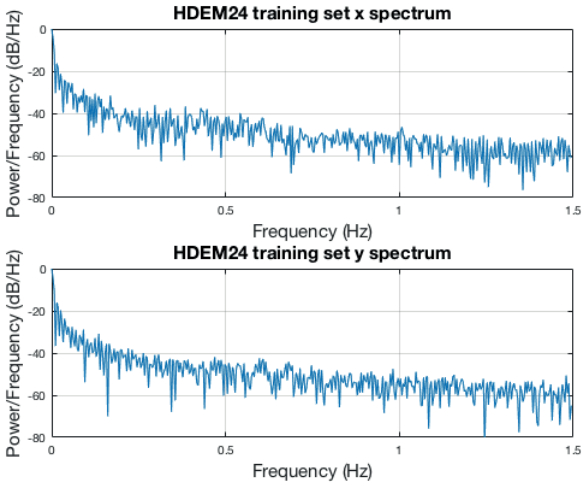


Figure 12 The frequency spectrum of the trajectory of the training set of HDEM24. Note the bandwidth of beyond -20dB is roughly 0.01 Hz. To focus on the low-frequency portion, only 1/5 of the horizontal frequency axis is shown.

Next, we apply the KCF tracker on this training video with the TFS ratio *N* varying from 2 to 100. The resulting AOR values subtracted from the AOR value without frame sub-sampling ($N = 1$) (ΔAOR) are plotted against the total computing time in Figure 13. The maximum performance enhancement occurs at $N = 10$. Other than $N = 79$, we observe that significant performance enhancement may still be achieved for $N > 10$. Our hypothesis is that for the testing videos, including the remaining video segment

of HDEM24, HDEM25, and CHPV72, similar performance enhancement and computing time reduction may be observed. We conduct the experiment and plot the corresponding ΔAOR in Figure 14. Indeed all ΔAOR values are positive in this range of *N*. However, the ranges vary for different video clips and the peak values occur at somewhat different values of *N* for different video clips. It appears *N* = 10 remains a reasonable choice. The take away here is that if the training videos are selected properly, the estimated frame sub-sampling rate may be applicable to other video clips of similar contents.

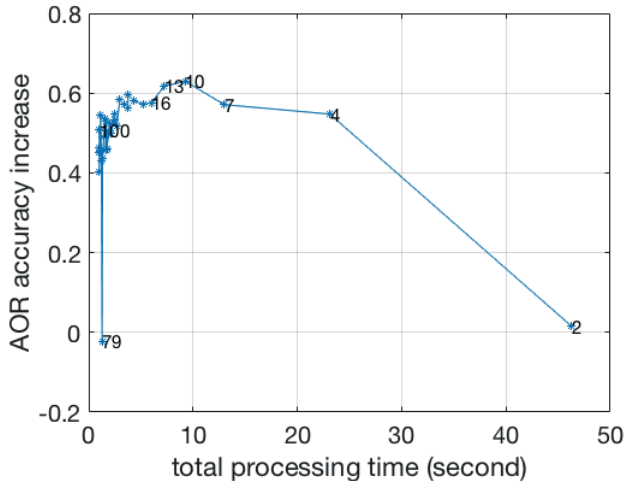


Figure 13 Accuracy increase of the training set with N=2 to 100

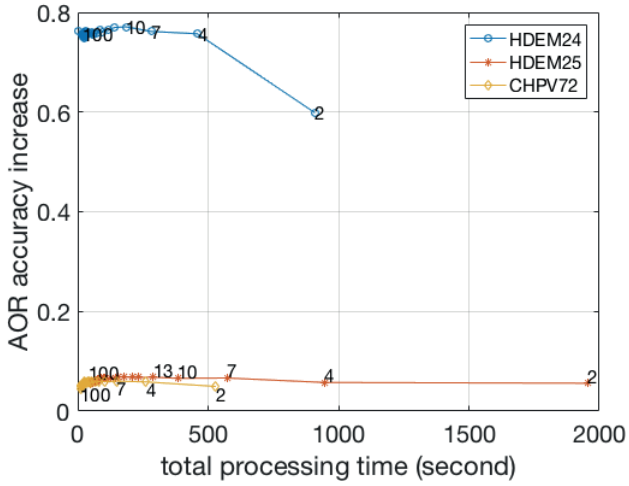


Figure 14 Accuracy increase of the testing data with $N = 2$ to 100. The AOR accuracy of N=1 for the 3 clips are: 0.145 (HDEM24), 0.908 (HDEM25), 0.924 (CHPV72)

3.1.2.3 Discussions

3.1.2.3.1 Rare Event Detection

In the discussion so far, we assume the AOR accuracy rate is the only performance metric for VOT tasks. However, very often, other performance metrics may be more desirable. For example, one possible use of the SHRP-2 video data set is to develop computer vision algorithms to detect rare events when the driver's head position is far away from the desired position, representing an event of distractive driving. If the longest duration of such an event is smaller than N frames (N is the frame-rate), then, it is possible that such an event will be completely be ignored when frame sub-sampling is applied, defeating the purpose of this video analytic task.

A remedy for this problem is to estimate the duration (in the number of consecutive frames) of such rare events. It would be desirable if the training sequence will be selected to contain such rare events. Then, the sub-sampling ratio N may be selected to be no more than half of the duration of the shortest event. Whenever such an outlier trajectory point is detected from the sub-sampled video sequence, the VOT algorithm may be applied locally around these frames to pinpoint the onset and offset of such an event.

3.1.2.3.2 Real-time VOT

For surveillance applications, real-time VOT processing is required. When an object of interest is detected, its trajectory is unknown to the VOT algorithm and hence frame sub-sampling may not be applicable initially. However, if the object to be tracked remains in the field of view of the camera, frame sub-sampling may be adaptively applied by increasing the value of *N* based on observed object trajectory so far.

3.1.2.3.3 Parallel Processing for Long Videos

When the video clips are very long, it is also possible to break entire video sequence into shorter sub-sequences and process each sub-sequence on a separate processor core to realize parallel processing and hence reduce the latency for processing the entire video. To do so, each video sub-sequence would need to be manually initialized. Alternatively, one may apply frame sub-sampling with a super large value of *N* (equal to the length of a sub-sequence) to automate the initiation task. A human operator then may confirm the initiation outcome more efficiently. Obviously, frame sub-sampling may still be applied to individual sub-sequences to realize further computing time reduction.

3.1.2.3.4 A Combination with Spatial Sub-sampling

Applying spatial sub-sampling of individual frames to speed-up VOT computation has been discussed in the literature [24]-[29]. However, an analytical error analysis on the VOT tracking outcome has not been provided. However, with the error estimation formula derived in section III, the impact of

spatial sub-sampling can easily be modeled as quantization errors imposed on the target position estimates. For example, if the search region is sub-sampled at a 2:1 ratio at each side, then the quantization error may be modeled as a uniform distribution over a range of [−1, +1] pixels at each of x and y dimensions. The variance of this random variable then will be added to that of the VOT tracking estimation error to yield the overall error estimate. Therefore, frame sub-sampling in the temporal domain can be combined with spatial sub-sampling in the spatial domain to realize computing benefit from both sides.

3.1.2.4 Conclusion

In this paper, we exploited the temporal correlation inherent in the video for VOT computing time reduction and an accuracy boost by frame sub-sampling. Criteria for assessing the applicability of VOT to a given video and the sub-sampling ratio are derived. A set of empirical rules for the proper application of frame sub-sampling to significantly reduce VOT processing time while enhancing VOT accuracy is also proposed. Applying these rules and criteria onto three hour-long SHRP 2 NDS videos, it proves the expectation that frame sub-sampling not only accelerates VOT computing speed but also improves tracking accuracy. This sub-sampling scheme could contribute a lot to saving computing time and resources and improving accuracy in video object tracking.

3.2 Drifting Resilience

Due to the variability of object appearance, it is often very difficult for the VOT algorithm to

determine if the current template used for tracking still contains the object. As such, the tracking algorithm may drift away from the target position, leading to tracking failure.

Many VOT algorithms made elaborate effort to avoid drifting. In the tracking-learning-detection algorithm [3], the detector corrects the tracker if necessary, and the learner estimates the detector's error. Babenko et al [4] used multiple instances of the object templates to reduce the template update error. In [6], a semi-supervised online boosting method is proposed to find a good model for the template update. In [33], a self-paced curriculum learning algorithm automatically selects "right" frames for the classifier to learn the templates, to ensure the updated model covers the right appearance. The long-term correlation tracking algorithm [34] trains an online random fern classifier to re-detect the object in case of tracking failure. STRUCK [5] uses kernelized structured output support vector machines to select and learn from targets online.

While trying to reduce the probability of drifting, these existing VOT methods do not explicitly address the issue of detecting the onset of drifting. As such, drifting remains an open problem in VOT [8]. The impacts of drifting exacerbate when VOT is applied to very long video clips containing hundreds of thousands of frames. If drifting occurs early without being detected in time, the remaining tracking outcome would be erroneous, wasting significant time and energy.

3.2.1 Drift Resilience: Drift Detection and Recovery

3.2.1.1 Drift Detection

The drift detection is based on the hypothesis that the quality (accuracy) of tracking can be estimated from the matching scores evaluated between the current template and candidate templates in different positions in the search region. When the maximum matching score falls below a learned threshold value, drift is likely to occur.

The matching score is an estimate of the likelihood probability $p(\mathbf{x}_k|\mathbf{z}_k)$ in Eq. (2.3) for each candidate object position $\mathbf{x}_k$ in the search region. Different VOT algorithms invoke different methods for estimating this matching score. However, most of them choose the position that yields that maximum value of the matching score as the maximum likelihood estimate of the object position at the k^{th} frame.

In the FSDR VOT algorithm, the predicted object state distribution $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ will be estimated, and the posterior probability for each candidate $\mathbf{x}_k$ will be evaluated using Eq. (2.3). Then Eq. (2.4) is applied to obtain the MAP estimate of the object position in the k^{th} frame. Meanwhile, the corresponding posterior probability

$$s_k \triangleq p\left(\hat{\mathbf{x}}_{k,MAP}\middle|\mathbf{z}_{1:k}\right) \tag{3.14}$$

will be used as a feature value to determine if drift occurs. Intuitively, if s_k is too small, the confidence of the current object state estimation will be very low, and it is likely drifting may occur.

Thus, the drift detection is solved as a hypothesis testing problem where drift is detected if $p(s_k|\mathrm{H}_1) > p(s_k|\mathrm{H}_0)$. H_0 and H_1 are respectively the null hypothesis (no drift) and alternate hypothesis (drift). These likelihood functions $p(s_k|\mathrm{H}_0)$ and $p(s_k|\mathrm{H}_1)$ may be learned empirically from the training video. Then a

threshold h_d may be chosen to balance between the probability of miss (failing to detect a drift) and false alarm rate (false detection of a drift). This is usually accomplished using a receiver-operating curve (ROC).

3.2.1.2 Drift Recovery

Upon detection of the onset of a drift event, the VOT algorithm can no longer rely on templates updated using recent frames. To recover from the drift event, the tracking algorithm must explore other means to recapture the lost object. These include back-tracking the algorithm to an earlier frame where confidence vector evaluates to a high confidence value of the estimated state.

Then, the tracking decisions may be adjusted in several ways:

a) Using one or more *known reliable* templates, usually including the initial template.

b) Using a generic object detector (e.g. a face detector, a pedestrian detector) if the class of the object is known as prior knowledge or can be classified using the initial template.

c) Ask for manual re-initialization.

C.-H. Chen et al. [35] have explored method a) and demonstrated its feasibility. For the SHRP-II NDS Driver's head tracking video sequence, the FSDR VOT algorithm employs a Viola-Jones (VJ) face detector [36] to re-capture the frontal face of the driver.

3.2.2 Sequential Bayesian Tracking Implementation

The FSDR VOT algorithm is based on the sequential Bayesian estimation framework described in section 2.1. Since it is developed for long video sequences, we assume a short training video sequence (< 3 minutes) is available to gather important statistics of the tracking algorithm. The following statistics are drawn from the training sequence HDEM25 over the first 2700 frames.

3.2.2.1 Prediction Phase

In Figure 15, we plot the distribution of 4-step displacement of the object (relating to the speed), $\mathbf{x}_k$-$\mathbf{x}_{k-1}$ for the video sequence sub-sampled at a ratio of 1:4. This is an estimation of the state transition probability $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ shifted to the origin.

The predicted state distribution $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ which specifies the search region is evaluated as

$$p\left(\mathbf{x}_k \left| \mathbf{z}_{1:k-1}\right.\right) = p\left(\mathbf{x}_k \left| \mathbf{x}_{k-1}\right.\right) + \hat{\mathbf{x}}_{k-1,MAP} \tag{3.15}$$

During the execution of the FSDR algorithm, $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ will be updated adaptively as new state estimates are computed.

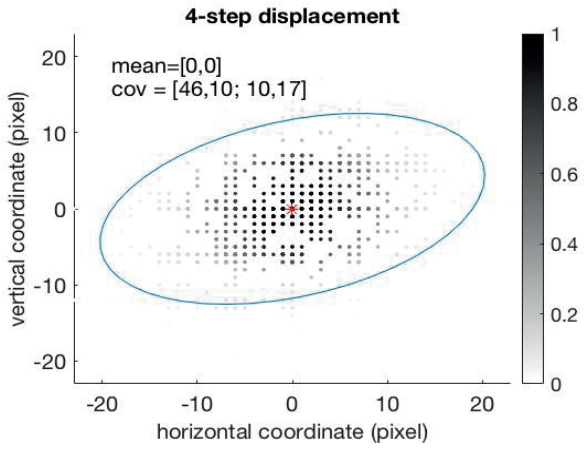


Figure 15 4-step head location displacement distribution

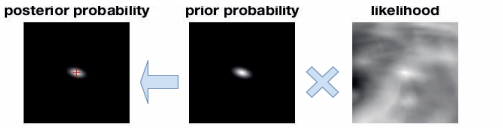


Figure 16 Probability score of each step of Bayesian estimation

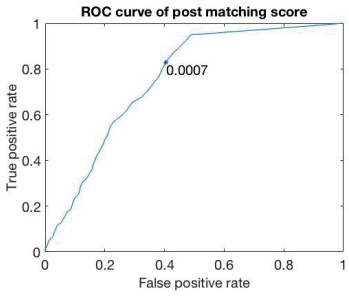


Figure 17 ROC curve for post matching score with the chosen threshold marked

3.2.2.2 Update Phase

At the k^{th} frame, a cross-correlation coefficient between the template and a rectangular candidate object template will be computed with its origin in the search region. The magnitude of this coefficient will be treated as an estimate of the likelihood $p(\mathbf{x}_k|\mathbf{z}_k)$ at the corresponding position. This probability

then will be multiplied by the prior probability $p(\mathbf{x}_k|\mathbf{z}_{1:k\text{-}1})$ estimated in the prediction phase to estimate the posterior probability $p(\mathbf{x}_k|\mathbf{z}_{1:k})$. This is performed for each pixel coordinates within the search region. A pictorial depiction of this step is shown in Figure 16.

Finally, $\mathbf{x}_{k,MAP}$ will be estimated according to Eq. (2.4). The corresponding confidence score s_k will also be computed using Eq. (3.14). And this triggers the drift detection module. To determine the threshold h_d such that a drift condition is determined if

$$s_k \le h_d \tag{3.16}$$

we collect $\{s_k\}$ for all frames in the training video, and use the center distance from the estimation to the ground truth to decide whether each s_k corresponds to drift or not. The empirical probability distributions of these two classes (H_0: no drift, H_1: drift) then leads to a ROC curve (cf. Figure 17) with each point on the ROC curve corresponding to a possible threshold value for h_d. We choose the value h_d = 0.0007 that corresponds to > 80% positive detection rate of the drift at about 40% false alarm rate.

If the result for the k^{th} frame is deemed no drift using the above drift detection threshold, the FSDR VOT algorithm moves to the next sub-sampled frame. Otherwise, drift recovery procedure will be invoked to produce a new template. With the newly estimated state vector, the object trajectory (state) of the N-1 skipped frames between k and k-1 sub-sampled frames will be estimated using linear interpolation:

$$\hat{\mathbf{x}}_{(k-1)N+n} = \frac{N-n}{N}\hat{\mathbf{x}}_{k-1,MAP} + \frac{n}{N}\hat{\mathbf{x}}_{k,MAP} \quad 1 \le n < N \tag{3.17}$$

A block diagram of the proposed FSDR tracking algorithm is shown in Figure 18.

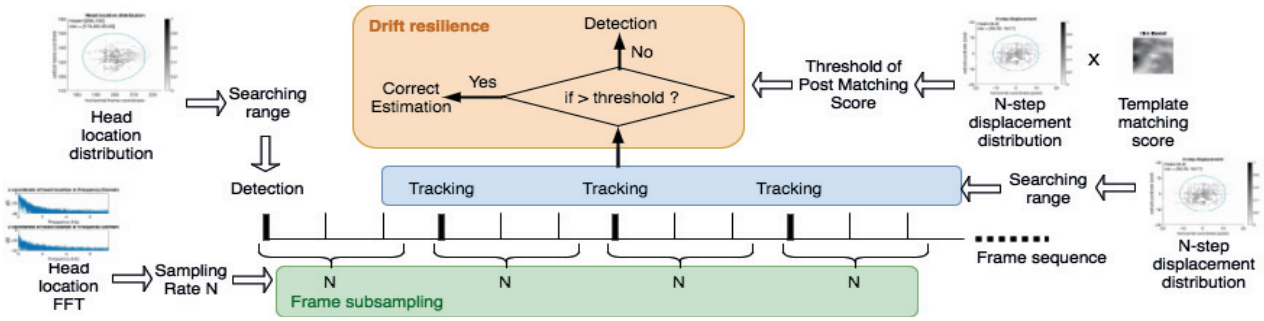


Figure 18 FSDR workflow

3.2.3 Experimental Results

Within the SHRP-II video, we used the first 2700 frames of HDEM25 for training and HDEM24 for testing. The performance criterion is the percentage of frames in the testing video that are accurately tracked. The cost function is the total CPU time for execution of the tracking algorithm. All the experiments are run on a MacBook Air with a processor of 1.3 GHz Intel Core i5.

To investigate the effect of sub-sampling at different sub-sampling factor *N*, we tried a sequence of N ranging from 1 (no sub-sample) to 25. The result for FSDR tracker is summarized in Figure 19. Two observations can be made: First, the computing time increases as *N* reduces. This is sort of expected. Secondly, the performance varies abruptly between adjacent values of sub-sampling factor *N*. But the performance gradually decreases as *N* increases.

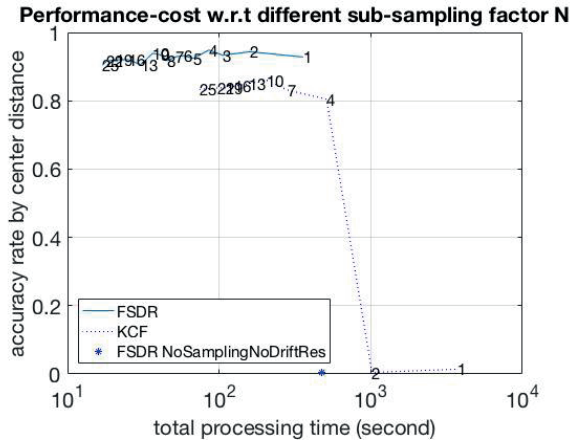


Figure 19 Performance-cost of FSDR and KCF

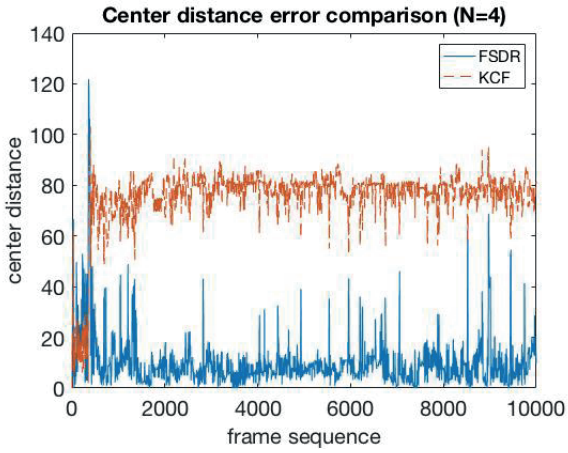


Figure 20 Center distance error comparison

We also ran FSDR but with no sub-sampling or drift-resilience. It is represented as a dot and located at the bottom of Figure 19. Its processing time is almost the same with FSDR with N = 1 but with drift-resilience. However, without drift-resilience, its accuracy drops to 0.0045. Most of the estimations failed because of an early drifting.

For comparison purpose, we choose KCF, an outstanding fast-speed tracker with high accuracy, which has been occupying the top spots of the VOT Challenge for years. It claims a speed of 172 fps on the OTB benchmark while most classical state-of-art trackers are getting a speed of about only thirtyish fps, and at the same time, it achieves a more accurate precision [20]. Since we are discussing processing long-term video clips with low cost but high accuracy, this KCF is prominent as a capable tracker.

We overlay the performance-cost curve of KCF on the same curve of FSDR. It can be seen that KCF, as a competitive tracker on processing this 60-minute video, has achieved an accuracy of higher than 0.8 with less than 1000 seconds processing time when sub-sampling factor N is no smaller than 4. But in general, FSDR has shortened the processing time by one order of magnitude compared with that of FSDR when considering corresponding sub-sampling factor N. Besides, FSDR has got higher accuracy than KCF by about 0.05.

It’s obvious that the curve of KCF drops when sub-sampling factor N decreases to 2 or 1. Figure 20 shows the center distance from the estimation position to the ground truth position of KCF and FSDR when N = 2. It shows an early drift happened to KCF while FSDR corrects all the drifts back on track.

3.3 Driver Behavior Analysis Tool

Besides the analysis of the driver's head, we also provide hands state detection and road state estimation into the driver's behavior monitoring. With the estimation of these factors, the driver's behavior analysis could become more comprehensive.

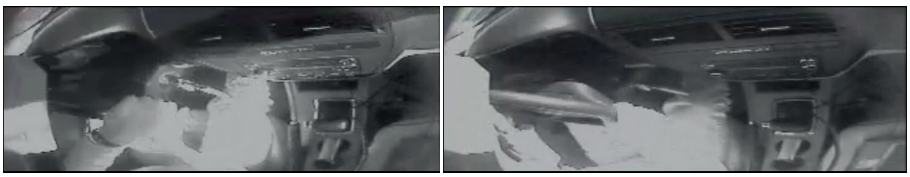
3.3.1 Hand State Detection

3.3.1.1 Overview of Hand Status Estimation

Driver's hands play an important role in analyzing the driver's state and vehicle safety. In the distraction estimation model, right hand and left hand on/off the steering wheel may indicate whether the driver is focused on driving. The assumptions are: (a) If at least one of the driver's hands is on the steering wheel, the driver is probably focused on driving. (b) If both of the driver's hands are off the wheel, it may indicate the driver is engaging distractive activities.

The main challenge in estimating the driver's hand status using the SHRP 2 video is the quality of the video. The hand status monitoring camera is mounted below the rear-view mirror, looking downward on the side view of the steering wheel. Hence, when the driver's hands are on top of the steering wheel, they are observable by this camera. However, when a hand is placed on the opposite side of the wheel (the lower part), the image of the hand is likely blocked by the steering wheel itself and perhaps the image of the right hand. Moreover, due to the complex background and inadequate lighting, it is difficult to distinguish the hand images from the background. Several examples of these challenging conditions are shown in Figure 21.

(a) (b)



(c) (d)

Figure 21 Sample challenging cases: (a) The left hand is holding the lower part of the steering wheel and thus its image is blocked by the steering wheel. (b) The right hand is holding the left edge of the steering wheel and thus it is blocked by the equipment between the camera and itself. (c) (d) The right hand can't be distinguished well from the background due to the extreme illumination.

3.3.1.2 Method

The video processing algorithm consists of the following: (a) manually label a straight line representing the side view of the steering wheel in the first frame, and (b) use the intensity variation on two parallel lines along the side view of the steering wheel over multiple frames to construct a temporal intensity variation map to decide whether the driver's hands are on the steering wheel.

Referring to Figure 22, the algorithm first requests the user to click the left and right ends of the steering wheel of the first frame of the video. Then a line is drawn between these two points, representing the location of the side view of the steering wheel. Next, a parallel line is drawn right below this first line, off the image of the steering wheel. If the driver's hand is placed in the lower part of the steering wheel, the image of that hand will cross over this second line. Examining the intensity along this second line allows one to infer whether the hand is holding the steering wheel from the bottom.

Nonetheless, this does not prevent the image of the other hand on top of the steering wheel blocking the view of the hand underneath the steering wheel, especially when the steering wheel is turned in a counter-clockwise direction.

The intensity variations on these two lines may reveal information of hand status. Along the upper line, the steering wheel usually has a uniform, darker intensity while the driver's hand looks much brighter. Hence, intensity increase over a certain segment of the line may indicate the presence of a hand on top of the steering wheel. On the bottom line, it may cross over both of driver's hands, on top or at the bottom of the steering wheel. Thus, making hand status estimation frame by frame requires additional efforts.

Fortunately, there is a high correlation between hand statuses in successive video frames. This motivated us to develop a new feature called a temporal intensity map as shown in Figure 23 to aid the hand status estimation. In a temporal intensity map, each row consists of the concatenation of the two parallel lines extracted in Figure 22 in a frame. The left-hand side is the intensity value along the top line, which covers the top part of the steering wheel. The right-hand side is the intensity value along the bottom parallel line, which may indicate if there are one or two hands underneath the steering wheel. The entire temporal intensity map consists of a stack of rows corresponding to successive frames. It can be seen that persistent holding of steering wheels appears as a bright vertical bar in this map. In particular, the bright vertical bar near the right end of the left panel indicates the right hand of the driver is on top of the steering wheel over the entire period. On the right panel, two vertical bars can be

observed. The left one indicates the left hand of the driver is at the bottom of the steering wheel because it does not show up in the left panel. The right vertical bar corresponds to the right hand that is on the top of the steering wheel. At the bottom of the figure, the sine-wave-like transition of these bars indicates turning of the steering wheel during that time.

Based on these observations, the algorithm computes the horizontal gradient of the map and applies a threshold to convert the map into a binary map for identification of the hand position. The binary map corresponding to Figure 23 is shown in Figure 24. If the width of any stripe in this binary map is larger than a preset threshold, then it is considered as a hand placing on (or beneath) the steering wheel.

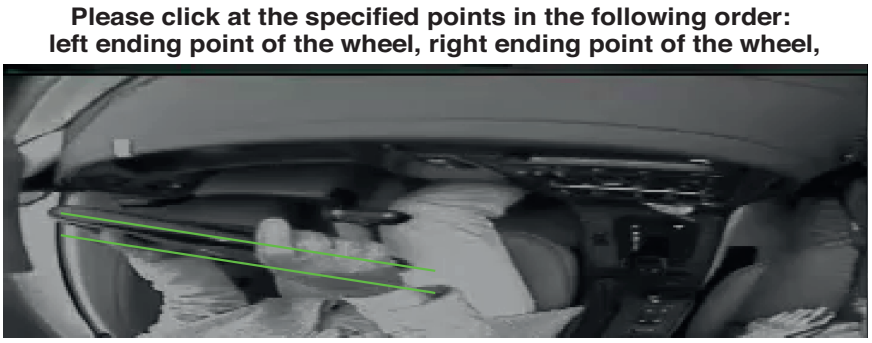


Figure 22 User interface of requesting a user to indicate the ending points of the steering wheel. The upper green line is drawn to connect the two points indicated by the user. The lower green line is generated automatically to be parallel to the upper line.

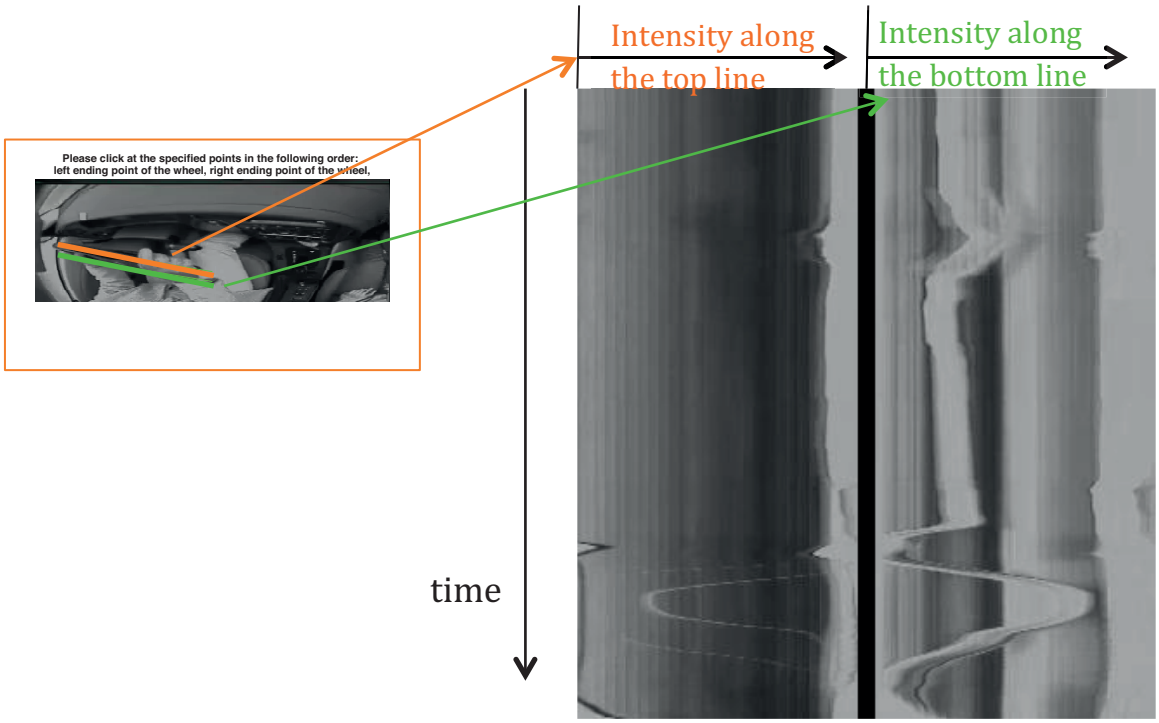


Figure 23 Generating an intensity map for a video clip

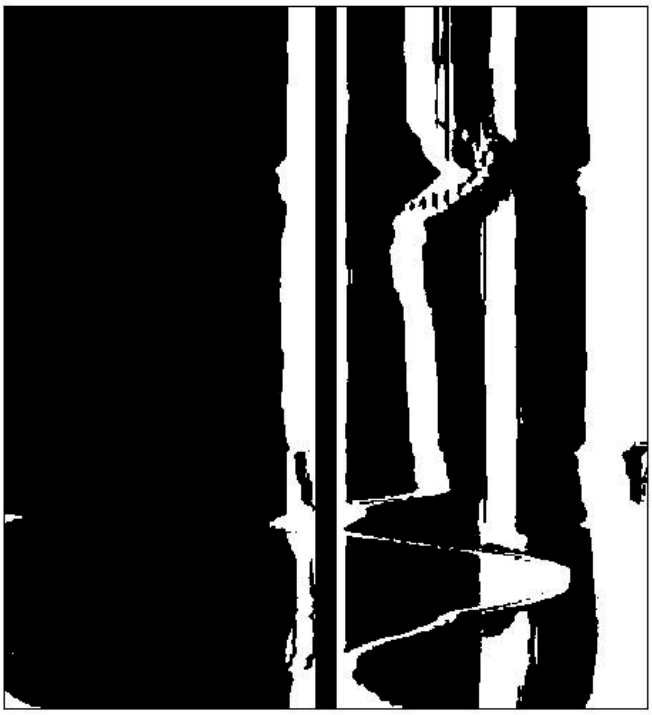


Figure 24 A binary map corresponding to the intensity map in Figure 23

3.3.1.3 Experiment Result

The outcome shows the prediction of whether at least one hand is on (or beneath) the steering wheel or neither hand is on the steering wheel. The experiment was conducted on 17 5-second video clips collected from SHRP2, and its detailed performance is summarized in Table 1. On average, the hand status (at least one hand or no hand on the steering wheel) of 93.74% of the frames are correctly estimated. However, to achieve this performance, the threshold of individual video clip must be adjusted manually and the algorithm is susceptible to noise.

Video Clip Name	Accuracy
hand_hdem26_OffOn.mp4	0.7567
hand_hdem26_OnOff.mp4	0.9733
hand_hdem26_OnOff2.mp4	0.8574
hand_hdem26_OnOff3.mp4	0.92
hand_hdem26_OnOffOn_GetTicket.mp4	0.92
hand_hdem26_swirling.mp4	0.8812
hand_hdem26_sunLight.mp4	0.8533
hand_s06nds_onoffwheel.mp4	0.8567

hand_s06nds_onwheel1.mp4	1
hand_s06nds_onwheel2.mp4	1
hand_s06nds_RHnotonwheel.mp4	0.9986
hand_s06nds_swirlingwheel.mp4	1
hand_cs06_handonwheel.mp4	1
hand_cs06_swirlingwheel.mp4	0.9688
hand_cs06_sunlight.mp4	1
hand_Video10Hands.mp4	0.9989
hand_Video10Hands.mp4	0.9966

Table 1 Hand status estimation result

3.3.2 Road State Detection

In SHRP2 NDS videos, a front-view camera mounted under the rear mirror will capture scenes in front of the car and hence record the road status. An example of a front camera video frame is shown in Figure 25. In the context of driving safety, the road state concerns the number of cars in front of the driving car being monitored within the field of view of the front camera. Since the camera is moving with the car, so does the background scenery, presenting more challenges to segment images of individual cars from the video frames.

Figure 25 An example of a front camera video frame

A few techniques detecting vehicles from images have been reported in the literature [37][42]. Two effective features for describing vehicles are the Haar wavelet [36] and Histogram of Oriented Gradient (HOG) [39] and classifiers used are AdaBoost [40] and Support vector machine (SVM) [41].

In this work, two road states are to be distinguished: (a) one or more vehicles appear in the video frame, versus (b) no vehicle appears in the video frame.

3.3.2.1 Algorithm

Step 1. Training Haar cascade classifier

A dataset consisting of 100 images of rear views of vehicles has been selected from Sandford and MIT image dataset [43][44]. Bounding boxes of individual vehicles are manually segmented. Besides, 1500 images with no vehicle inside have been collected from SHRP2 video to serve as negative instances. A cascade AdaBoost classifier using Haar wavelet features is trained using function

opencv_traincascade provided by OpenCV library. Since the targets in positive instances are all rear view of vehicles, this detector is only trained for detecting the rear view of vehicles, which fits the purpose of detecting vehicles from the SHRP2 video recorded by the front-view camera.

Step 2. Detect vehicles

After the vehicle detector has been trained, the video will be processed with lane detection, vehicle detection, false-positive elimination, and traffic status estimation. The detailed steps are as following:

1. Extract lower half of each frame.

2. Apply the Canny edge detector to detect edges, and then apply Hough transform to find lanes.

3. Since many straight lines may be found in the same frame, the detected lines are grouped into the right side (blue line) and the left side (red line). And for each side, an estimated lane (pink line) will be calculated based on lines detected in step 2 (Sample result is shown in Figure 26). Its parameter k and b (the line is expressed by y = k*x +b) will be recorded.

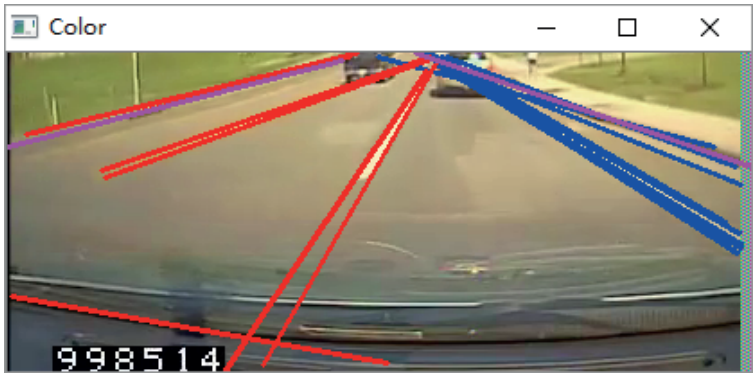


Figure 26 Lane detection in the lower half frame

4. Detect vehicles on this lower half-frame using the trained Haar cascade classifier.

5. Record estimated lanes and detected vehicles of each frame into a CSV file.

6. Remove false detections by applying following rules

    a) Spatial inconsistency: If the location of the detected vehicle is too far from the detected lanes, then it is a false detection.

    b) Temporal inconsistency: Over a window of 10 consecutive frames, if a vehicle is detected within a circle with a radius of 60 pixels in fewer than 5 frames, then it is likely to be a false detection.

    c) If two vehicles are detected in the same frame with positions separated within 20 pixels, one of them is likely a false detection.

3.3.2.2 Discussion

Sayanan Sivaraman et al. [38] reported a vehicle detection system called ALVeRT that uses a video dataset named LISA containing 3 clips: Dense, Urban, and Sunny. Each name represents the climate when its video was recorded. The research team compares the outcome of above-mentioned TrafficDensity_Detector against those reported in [38]. The results are summarized in Table 2 where the True Positive Rate (TPR) and the False Discovery Rate (FDR) are reported. Our TrafficDensity_Detector performs better in Dense and Urban cases (higher TPR and lower FDR) and on a par with or slightly inferior to ALVeRT in sunny.

LISA	Tracking System	TPR	FDR
Dense	ALVeRT	89.5%	51.1%
	TrafficDensity_Detector	94.33%	32.37%
Urban	ALVeRT	83.5%	79.7%
	TrafficDensity_Detector	99.67%	19.19%
Sunny	ALVeRT	98.1%	45.8%
	TrafficDensity_Detector	98.33%	55.24%

Table 2 Vehicle detection comparison

3.3.2.3 Experiment

The research team collected 32 clips where at least one vehicle is around (name it as a vehicle-around dataset) and 46 clips where the road is empty (name it as a non-vehicle dataset). All of these clips are 5-second clips from SHRP2 videos. All are recorded in the daytime, and a majority of them are in cloudy weather, while some are in sunny weather.

The total number of detections for each video has been counted. And for the vehicle-around and non-vehicle dataset, the histogram of them has been plotted separately in Figure 27. The figure shows that for most of the non-vehicle clips, the total number of vehicle detection is 0, but several clips have got false positive detections. As for the vehicle-around clips, most of them have got a number of detections while 3 clips have got false negative detections with less than 2 vehicles detected. By setting

the threshold of the total number of vehicles detection per clip, an ROC curve can be plotted. In consideration of saving computation load and time consumption, sampling rate from one detection per 75 frames to one detection per frame for vehicle detection is applied. And their ROC curves are plotted in Figure 28. Since one detection per several frames can't have consistent detections, only the one detection per frame trial has got consistency criteria applied.

The ROC curve with the best performance, which has the largest area under the curve, is one detection per frame and using consistency to eliminate false positives. And the threshold corresponding to the best performance is 6.67 detection per clip. So if a 5-second clip gets the number of detection more than 6.67, then this clip is classified as representing a crowded traffic scene. While if this clip has got less than 6.67 vehicle detection, this clip is determined as showing an empty street. Its confusion matrix is shown in Table 3(a).

A testing set is collected containing 10 vehicle-around clips and 10 non-vehicle clips. All of them are 5-second clips from SHRP2. The one detection per frame scheme is applied and the threshold of 6.67 is applied. The confusion matrix is shown in Table 3(b). It shows that this approach can indicate traffic density.

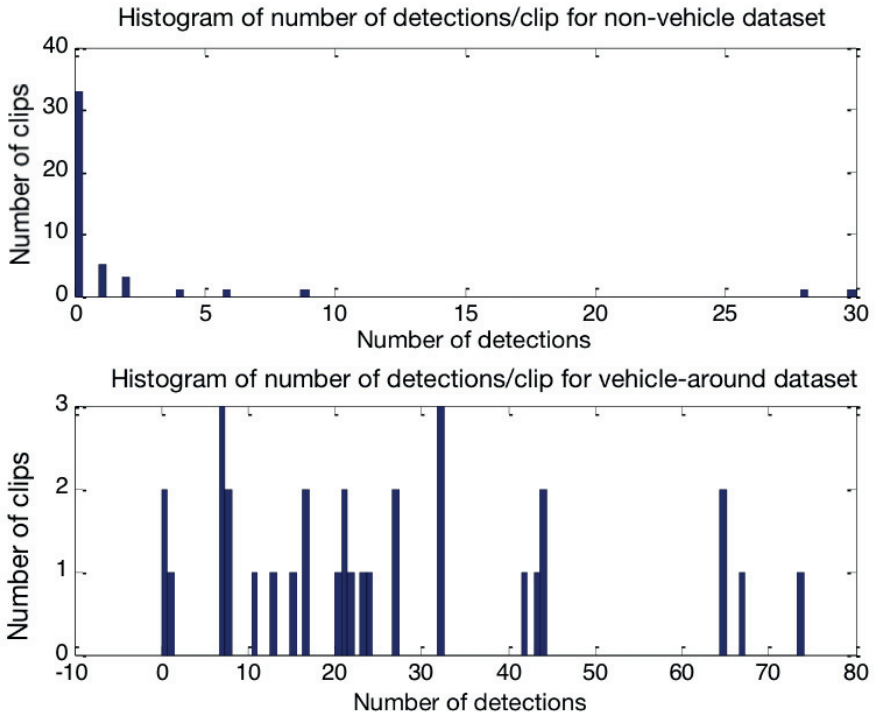


Figure 27 Histogram of number of vehicle detections per clip

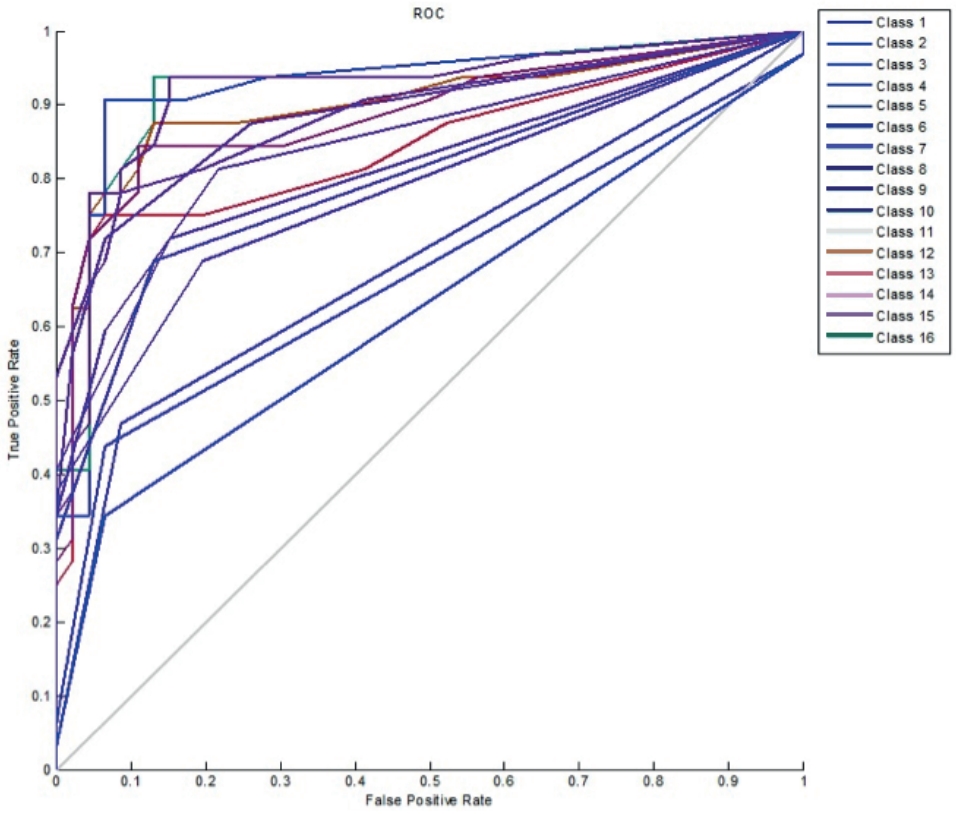


Figure 28 ROC curves of different sampling rates

Prediction / Ground Truth	Vehicle-around	Non-vehicle
Vehicle-around	90.63%	9.37%
Non-vehicle	6.52%	93.48%

(a) Confusion matrix of the training dataset

Prediction / Ground Truth	Vehicle-around	Non-vehicle
Vehicle-around	80%	20%

Non-vehicle	10%	90%

(b) Confusion matrix of the testing dataset

Table 3 Confusion matrix of classifying traffic status

3.3.2.4 Conclusion

From the confusion matrix, it can be seen that this road state estimation algorithm provides a good traffic density prediction.

However, the sampling frequency can't be sacrificed to save the computation load. Although applying vehicle detection on every frame consumes a large amount of time, it is most reliable when the detection is performed on every frame.

Besides, the quality of the vehicle detector is crucial to traffic density estimation. Though the detector trained in this project has achieved a competing performance with the initial vehicle detector in [38], a larger-sized vehicle-image dataset containing more types of vehicles will help to provide a better vehicle classifier with fewer false positives but detects out more vehicles.

Chapter 4. Video-based Lifting Risks Analysis

4.1 Abstract

A widely used risk prediction tool, the revised NIOSH lifting equation (RNLE) provides the recommended weight limit (RWL) for two-handed lifts, but it is limited by analyst subjectivity, experience, and resources. This chapter describes a robust, non-intrusive, straightforward approach to automatically extract spatial and temporal factors necessary for applying the RNLE using a single video camera view of the sagittal plane. The subject's silhouette is segmented by motion information and the novel use of a ghosting effect provides accurate detection of lifting instances, and hand and feet location prediction. Laboratory tests using 6 participants performing 36 lifts each (N=216) showed that a 640 ×480 pixel 2D video in comparison to 3D motion capture, provided RWL estimations with an accuracy of 0.2 kg (SD = 1.0 kg). The linear regression between 2D video estimated and laboratory motion tracking RWL was $R^2 = 0.96$ with a slope and intercept of 1.0 and 0.2 kg, respectively. Although the current study used low definition video in order to synchronize with 3D motion capture, better performance is anticipated using high definition video. This novel method makes automatically evaluating the RNLE practical for implementation on a smartphone or handheld device.

4.2 Lifting Monitor Algorithm

The lifting monitor algorithm takes advantage of motion information to distinguish the moving subject from the static background. Based on the spatial and temporal features of the video scene, a

ghost effect is exploited to detect the lifting instance and hand locations. A rectangular bounding box drawn around the subject is used to locate the feet.

The algorithm contains three steps: (1) moving target detection, (2) action feature extraction, and (3) feature recognition, as shown in Figure 29. Each step is illustrated in Figure 30.

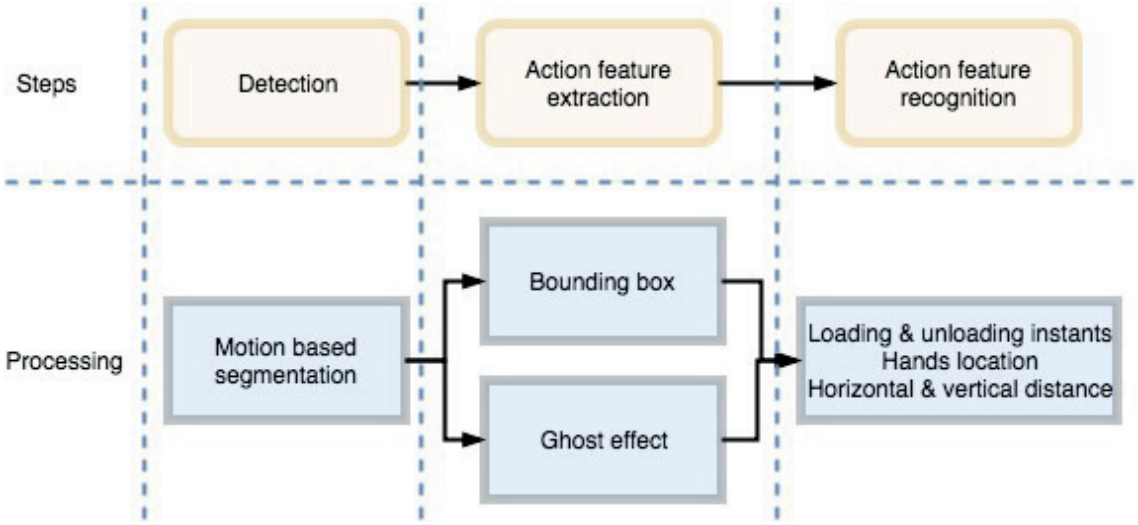


Figure 29 Flowchart of lifting monitoring algorithm

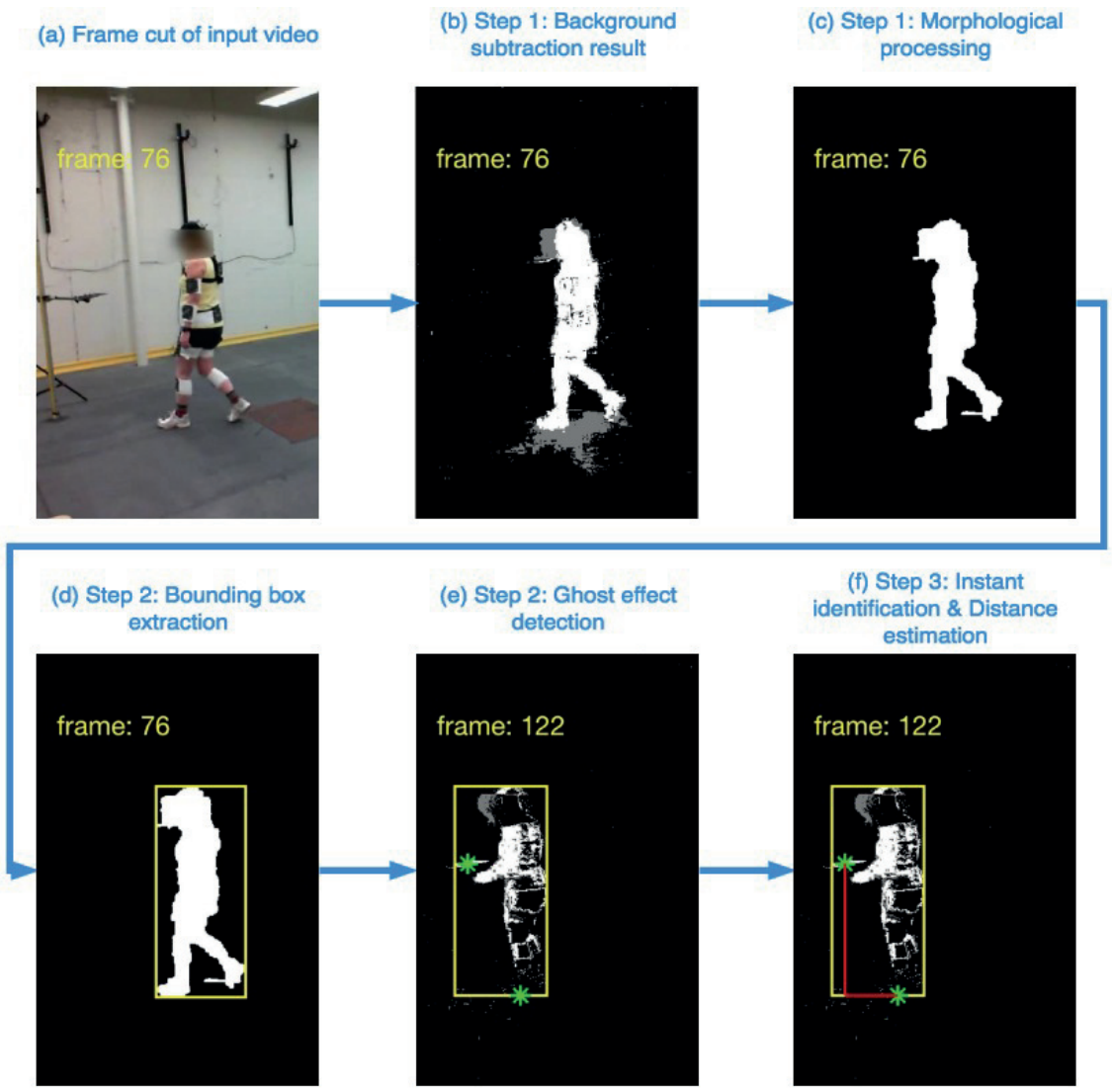


Figure 30 A demonstration of the video processing methodology for each step of the algorithm. Images (a), (b), (c) and (d) show the sequence of procedures for the 76th frame of a video where the subject walks to the lift location in the 122nd frame. Images (e) and (f) demonstrate hand and ankle detection at the 122nd frame of the video where the subject is lifting the object. A rectangular bounding box encloses the worker. The stars show the detected location of the hands and ankles. The lines connecting the stars indicate the horizontal distance from hands to the ankles and the vertical distance from the hands to the ground, respectively.

The moving target detection step leverages the fact that the subject moves during lifting while most of the surrounding environment is static, and thus a motion-based technique could segment out the moving items. In this algorithm, a mixture-of-Gaussian background subtraction algorithm ([69], [70]), known as MOG2, is applied. The background subtraction algorithm builds a background model for each pixel using a mixture of Gaussian distributions and updates the weights of the texture to represent the time proportions that the pixel colors remain in the scene. The result is an $M \times N$ matrix map, where M and N is the height and width of the image in the unit of pixels, labeling each pixel as foreground (moving), background (static), or shadow, with respective colors white, black and gray. Thus, the detected foreground can be identified pixel-wise by the white-colored foreground mask. Each of the connected clusters of the white pixels is referred to as a blob. In a working environment where only the subject moves, the resulting map would capture the subject perfectly with a foreground mask identifying the subject's silhouette. But, several moving objects and light reflections could introduce noise (false positives), and the foreground mask of the subject might be incomplete and distorted (false negatives), as shown in Figure 30(b). These false detections are due to a false match to the background model. False detections are eliminated by morphological operations based on their size relative to the larger size of the subject's silhouette, as shown in Figure 30 (c).

The action feature extraction step utilizes information from the first step. The height and width of the subject's silhouette are extracted, represented by a rectangular bounding box tightly covering the

subject's foreground mask (Figure 30 (d)). Another feature utilized is the condition of the background subtraction, which is a set of connected points detected as in-motion, but not corresponding to any real moving object, named the ghost effect. This phenomenon appears when an object is moved away or set to a location, changing the appearance of an area (Figure 30 (e)). It persists for a short period, depending on how fast the background model absorbs the new appearance of that area. The ghost effect is detected in accordance with these four properties:

(1) Consistency in time. A blob of the ghost effect area should show up in the same location in subsequent N frames.

(2) Gradual vanishing. The size of the blob should be no larger than the size of the lifted object and should gradually get smaller.

(3) Proximity. When the ghost effect shows up, the subject's silhouette (large-sized blob) should be close to it. In a non-ideal environment where there is more than one moving object, the ghost effect might occur in multiple places. We, therefore, use proximity to focus the ghost effect caused by the subject.

(4) Frame number. The frame number N is the duration that the ghost blob persists when the background model updates the new appearance of the ghost effect area into the background model.

The feature recognition step exploits the fact that the ghost effect occurs when the appearance in a location changes when a stationary object is moved by the subject, and, therefore, the loading instance

can be identified from the start of a ghost effect. At these detected lifting instances, the respective location of each ghost effect indicates where the hands were, and the bottom blobs of the silhouette show where the feet were during the initiation of the lift. If the subject is recorded in the view from the sagittal plane, and the hands and feet are moving symmetrically about the sagittal plane, the ghost blob center can be used as an approximation of hands center during the initiation of the lift. Similarly, when an object stops moving such as when setting down an object at the termination of a lift, the ghost effect occurs and persists indicating the instance of release.

The geometric center of the bottom portion of the subject silhouette blob (i.e. the lower 10% region of the bounding box while the subject is standing) of the subject can be used to approximate the horizontal location of the midpoint of the subject's ankles. Thus, the horizontal distance H from hands center to ankles (for the HM calculation) and the vertical distance V from hands to the ground (bottom edge of the bounding box for VM and DM calculations) can be estimated (Figure 30 (f)). The loading and unloading instances are important for H and V distance estimation because they indicate when to detect the hand and ankle locations. The algorithm considers the hands as the center by the ghost blob, which does not move in time. Distances measured in pixels were calibrated against the subject’s standing height.

4.3 Validation of the Algorithm

4.3.1 Laboratory Data

Data utilized in the current study were from a previous study conducted by researchers at the National Institute for Occupational Safety and Health (NIOSH). The aim of the original study was to record symmetrical lifting tasks using the combination of two lifting task variables (horizontal and vertical distances) defined by the American Conference of Governmental Industrial Hygienists (ACGIH) Threshold Limit Values (TLV) for lifting (ACGIH, 2007). The TLV for lifting classifies 12 risk zones using the two task variables for symmetrical lifting in the sagittal plane. The horizontal distance (H) is defined as the projected distance on the transverse plane from the centre of two ankles to the centre of two hands. The vertical distance (V) is defined as the distance from the centre of two hands to the ground. The midpoints of risk zones were used as the positions for starting the lifting tasks, except for Zones 1-3, 4, 7, and 10. The alternative starting locations of the lift tasks for the exceptional zones were chosen for realistic lifting motion within each subject's reach envelope. The origins of the 12 lifting tasks in relation to the subject's neutral body position are shown in Figure 31. Each task was repeated three times for a total of 36 lifting trials for each subject. These trials were assigned to each subject in a random order.

A wired grid weighing 0.45 kg and measuring 36 × 12 cm, with two cut-out handles, was used to simulate lifting a tote box during the trials. The grid was designed to help subjects create realistic lifting motions while minimizing obstructions for motion tracking. The grid was set on a small platform (12 ×

12 cm) for setting the initial lifting height. The platform was connected to a movable clamp on a metal pole for adjusting the lifting height. The small platform provided clearance for subjects to lift the grid in a natural motion. Subjects aligned their toes against one of three marked lines in the lifting area to create three horizontal distances for three grouped risk zones (Group 1: Risk Zones 1,4,7 and 10; Group 2: Risk Zones 2,5,8 and 11; Group 3: Risk Zones 3,6,9,12). These lines were 25.4, 45.7 and 71.1 cm from the centre of the grid (i.e., the centre of two hands) to the centre of their ankles.

Subjects were asked to walk from a marked line (i.e. initial position) to the lifting area and line up their toes against one of the marked lines for the lifting task. The vertical height of the grid for the lifting task was set up prior to each trial. The distance between the initial position and the starting point of the lift task typically required subjects to take 3-4 steps prior to lifting the grid. After lifting the grid, subjects were asked to turn around and continue to carry the grid and set it down on a shelf in a fixed height of 77.5 cm. After setting down the grid, they were asked to turn around and walk to a marked finish line for completing each trial. The distance between the shelf and the finish line typically required subjects to take three to four step to complete the trial. Subjects were instructed to walk and lift/carry the grid with two hands at their own pace and in their preference of direction for the two turnarounds. They were also instructed to carry the grid in front of their body to minimize trunk asymmetry. Each trial was completed continuously in about 15 seconds. Subjects practiced a few times until they familiarized themselves with the entire experimental procedure.

The video data for each trial were recorded by a web camera (Microsoft 1080p LifeCam) in

synchronization with whole body motion data measured by a motion capture system (Optritrack 12 IR camera system, model Flex 13 with the MotionMonitor data acquisition program, Innovative Sports, Inc., Chicago, USA). The web camera was set up in a fixed location at the typical eye level (165 cm from ground) and approximately 3.92 m in distance from the beginning of each trial. The camera viewing angle (88°) to the subjects' initial standing point was near perpendicular to the direction of the trial. The resolution of the video recordings was set at 640 × 480 pixels at a 30 fps rate to meet the hardware synchronization requirements for the motion capture system.

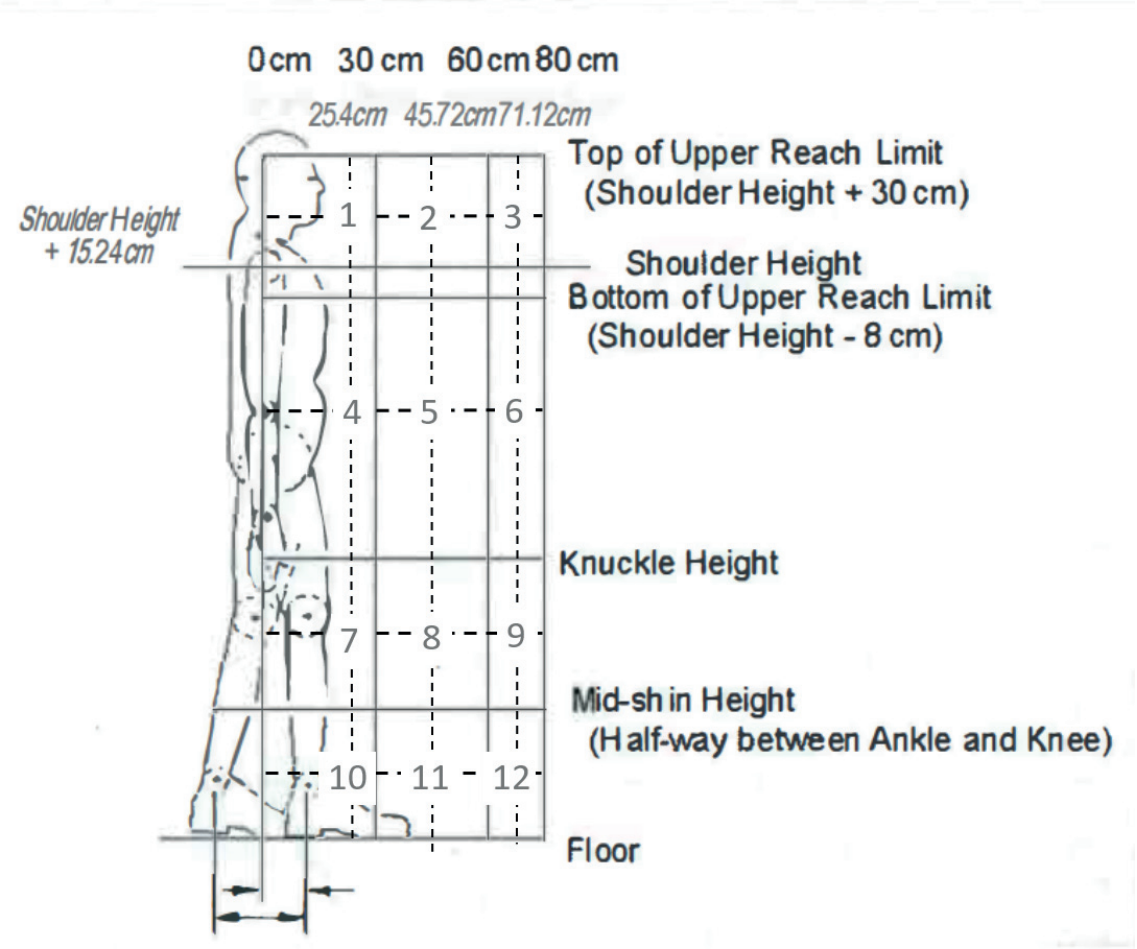


Figure 31 The starting point of 12 different lifting tasks was designed using the 12 risk zones of the ACGIH TLV for Lifting. The intersections of the dotted lines are the origins of the tasks and most were in the centre of the risk zones. The vertical heights for Tasks 1-3 were adjusted to 12.24 cm above the subjects' shoulder height. The horizontal distances for tasks 1, 4, 7 and 10 were adjusted to 15.4 cm from the centre of the two ankles. These adjustments were made to create a realistic lifting motion. Adapted from ACGIH_TLV lifting risk zone system.

Motion tracking marker clusters were attached to 13 body segments of the subjects for tracking their whole-body motion by the MotionMonitor program using 12 IR cameras. The body locations for marker clusters and five inertial moment unit (IMU) wearable sensors (Kinetic Inc.) are shown in Figure 32. The (IMU sensors were for a previous study and not used in the current study). The motion capture system was calibrated according to the standard operating procedure of the OptiTrack company to achieve an average 0.7 mm accuracy across ten test sessions for motion measurements in the three-dimensional space.

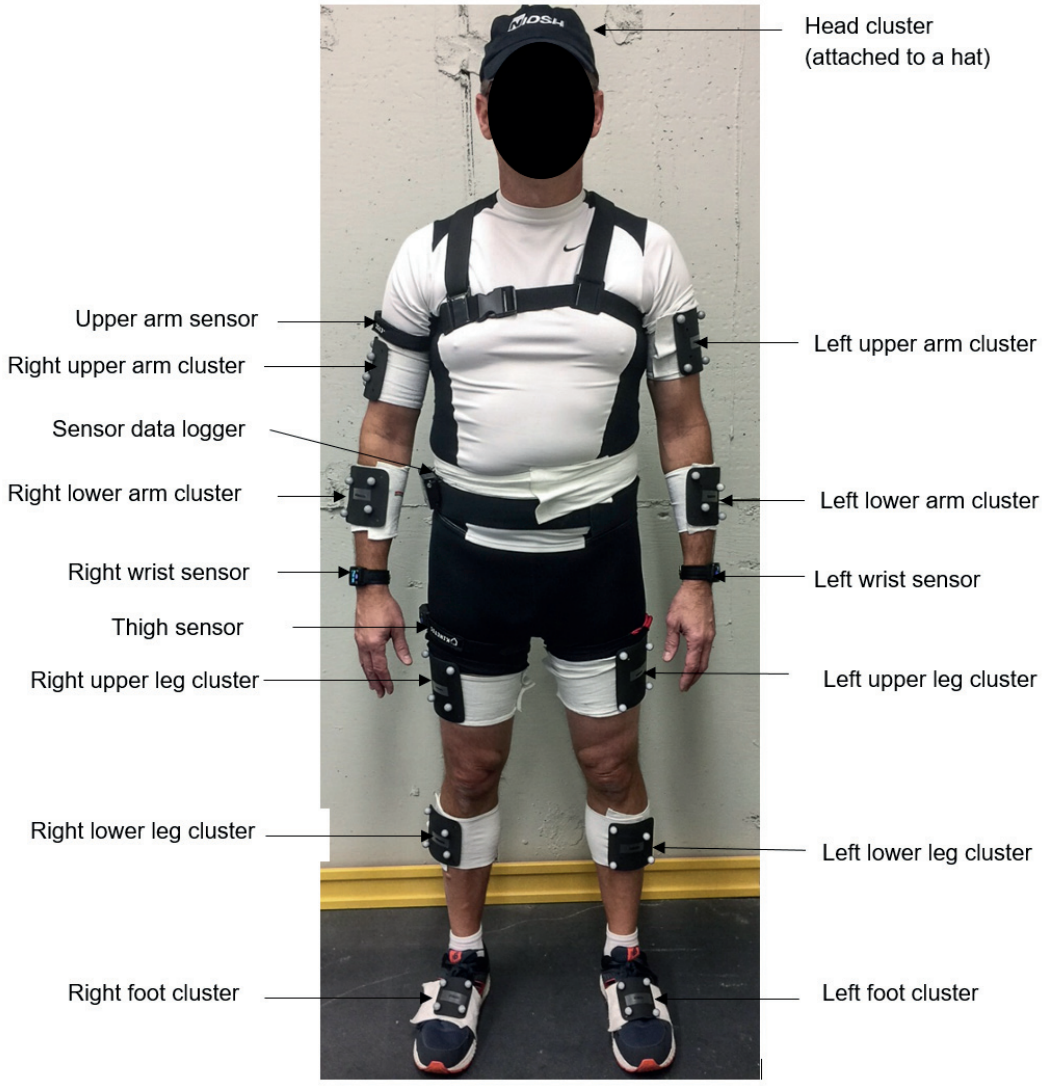


Figure 32 Demonstration of wearable sensor and marker cluster attachments to the body landmarks for the motion capture system. Each cluster has four small retro reflective Styrofoam spheres geometrically configured to be different from one another for motion measurements. Two clusters (upper and lower back) and one wearable sensor attached to the back of the chest Velcro assist harness are invisible in this picture. White elastic Velcro straps are used for the clusters, and thinner black elastic Velcro straps are used for the upper arm and thigh wearable sensors. Two wrist sensors are attached by adjustable rubber bands.

Data acquired and derived from the motion capture system were used as the ground truth data for evaluating the accuracy of the video lifting monitoring algorithm. Three variables were used for validation: the time instance at the beginning of the lift (BOL), the time instance when the plate was set down at the end of the lift (EOL) as well as H and V for the BOL. Two researchers reviewed video recordings of the trials in the MotionMonitor program and manually recorded the video frame numbers to establish the BOL and EOL. The criterion for determining the frame numbers was based on the moment when the grid started to move for the BOL with two hands. Each researcher independently reviewed half of the trials, but when in doubt, they discussed questionable frame numbers to reach agreement on the final value. Once the video frames for the BOL were determined, the data were used to identify the H and V variables calculated with the motion capture data. A similar procedure was used to determine the EOL when the grid was initially set down.

4.3.2 Subjects

Six subjects were used for this study. They were recruited among employees in the division of the Applied Research and Technology office of NIOSH in Cincinnati, Ohio. Prior to data collection, written consent was obtained from the subjects using the NIOSH-approved IRB study protocol. The subject inclusion criteria were individuals that were capable of (1) lifting a 1.4 kg weight in a location combining different horizontal distances from their body to the load within their reach and vertical distances from the shin to shoulder height, (2) lifting and carrying the 1.4 kg weight 3 m, and (3)

repeating 12 different tasks (described previously) 3 times for a total of 36 lifting trials. Exclusion criteria were individuals with musculoskeletal disorders or any pain at the time of recruitment or in the past three months, individuals under age 18, and being pregnant. Demographic and measured anthropometric data relevant to the study are provided in Table 4.

Table 4 Demographics and anthropometric properties (Mean ± SD) of study subjects (Male: N=3; Female: N=3).

Gender	Weight (kg)	Height (cm)	Age	Forearm length (cm)	Upper arm length (cm)	Thigh Length (cm)
M	101.8 ± 14.65	176.28 ± 1.39	55 ± 1.87	27.69 ± 1.39	32.26 ± 1.45	44.56 ± 3.46
F	69.66 ± 7.39	163.58 ± 4.61	48 ± 13.55	26.42 ± 1.66	30.73 ± 1.66	44.31 ± 4.52
Average	82.53	169.33	51.33	27.09	31.54	43.27
SD	18.94	9.04	11.50	1.91	2.03	3.87

The lifting monitoring algorithm was applied using the validation dataset (6 subjects with 36 clips each). Given a video clip as input, the algorithm automatically outputs the bounding box of the subject for each frame, the detected loading and unloading instance, hands location and feet location at the loading and unloading instance, and the RWL.

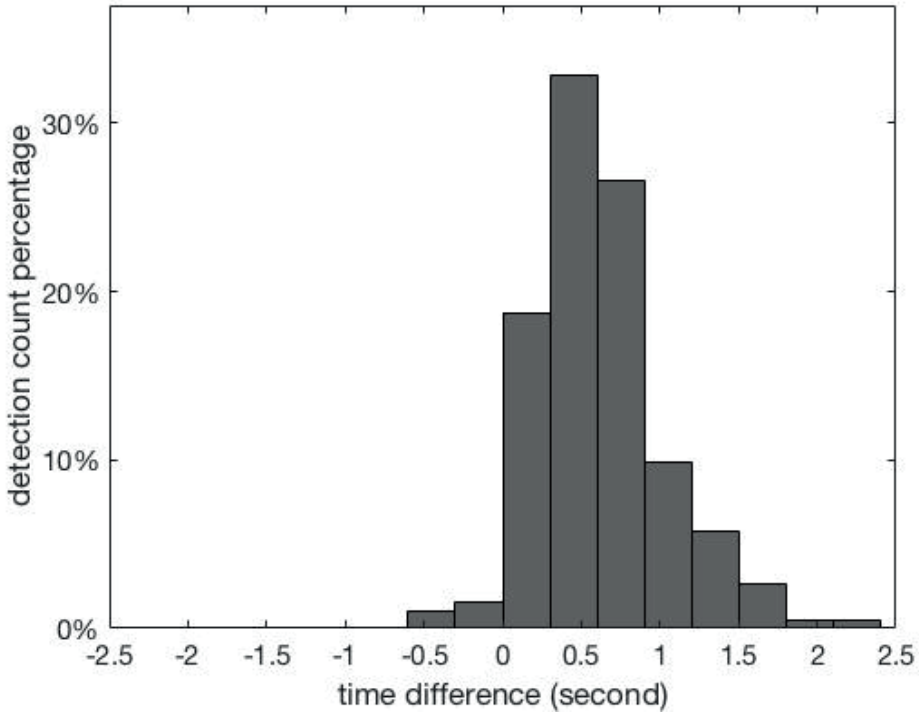
The NIOSH study was designed to focus on the origin of the lift in 12 lift zones. The RWL for the loading instance at the BOL was calculated, including the horizontal distance from the hands center to the ankles center (H), and the vertical distance from the hands center to the ground (V).

4.4 Results

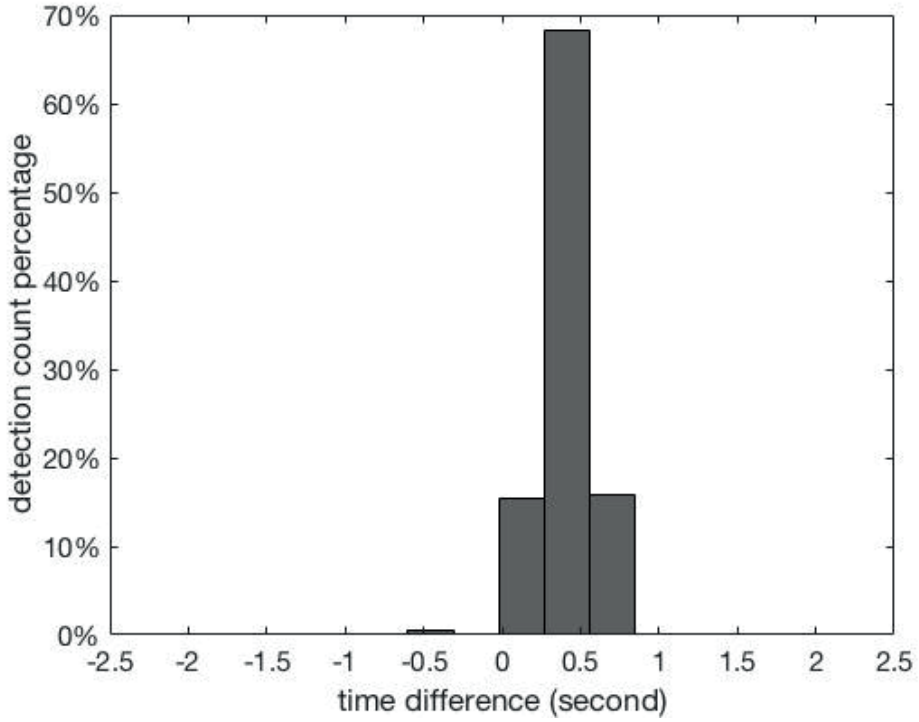
4.4.1 Lifting and Releasing Instance Detection

Since the ankles during loading remained steady for a short period, when the predicted lifting instance and the ground truth instance coincided with this period, the predicted lifting instance was considered successful. In this validation experiment, the lifting instance estimation was successful for all clips. The predicted lifting instance timestamp was compared with the manually observed instance (ground truth). The time difference between prediction and ground truth was calculated.

A histogram of the time difference for the BOL is plotted in Figure 33 (a). The mean time difference was 0.624 s (SD = 0.42 s). Based on this distribution, 99.5 percent of the measured lifting instance detections were within [-0.636, 1.884] s of the ground truth measure. A similar time difference for the EOL histogram for the time difference is plotted in Figure 33 (b). The mean time difference was 0.144 s (SD=1.52 s). According to this distribution, 99.5 percent of the measured unloading instance detections were within [0.125, 0.833] s of the ground truth measure.


detection count percentage
30%
20%
10%
0%
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
time difference (second)

A


detection count percentage
70%
60%
50%
40%
30%
20%
10%
0%
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
time difference (second)

B

Figure 33 Histogram for the time difference between ground truth and (a) the beginning of lift instance detection time, and (b) the end of lift detection time.

4.4.2 H Detection at the Loading Instance

The horizontal distance from the hands to the ankles (H) is a key factor in calculating the RWL. We reference the H data from the 3D motion capture system as ground truth and calculate the H difference between the video estimated H and ground truth H. A histogram for the H difference is plotted in Figure 34. The mean of the H difference was -1.17 cm (SD = 4.74 cm). Based on this distribution, 67 percent of the measured lifting instance detections were within ± 5 cm of the ground truth measure.

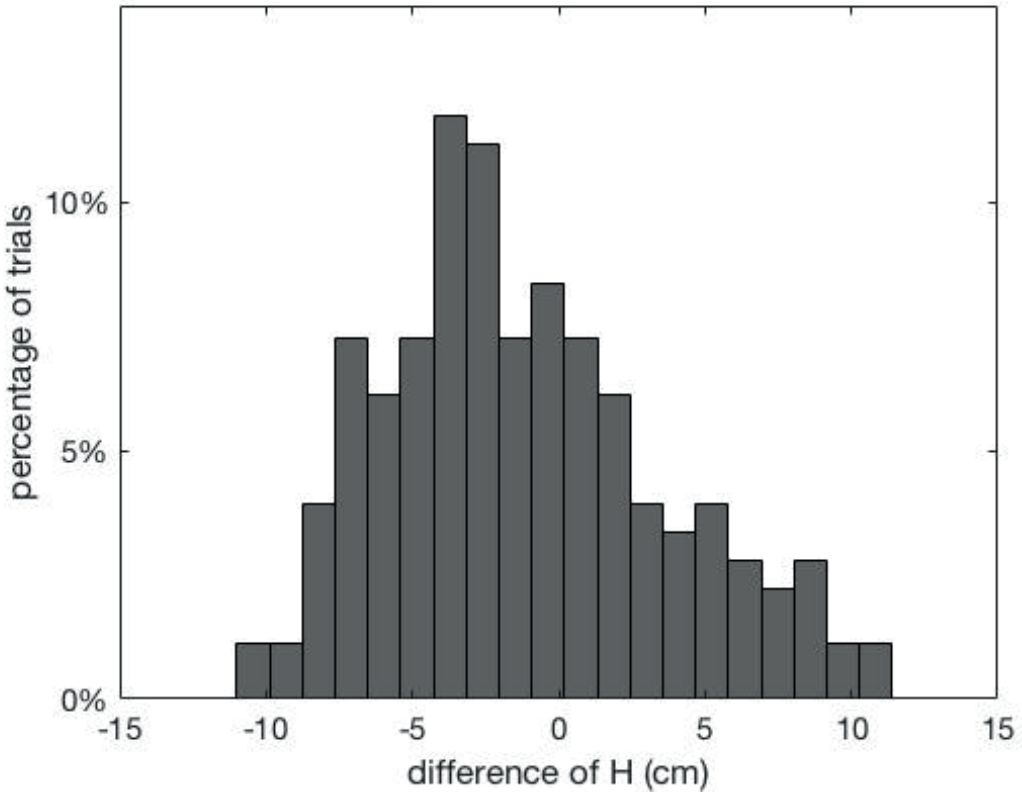


Figure 34 Histogram of the difference of H between motion capture and video detection at the loading

4.4.3 V Detection at the Loading Instance

The vertical distance from the hands to the floor (V) is another key factor in calculating the RWL. We reference the V data from the 3D motion capture system as ground truth and calculate the V difference between estimated and ground truth V. A histogram of the V difference is plotted in Figure 35. The mean of the V difference was -0.36 cm (SD = 3.22 cm). Based on this distribution, 88 percent of the measured lifting instance detections were within ± 5 cm of the ground truth measure.

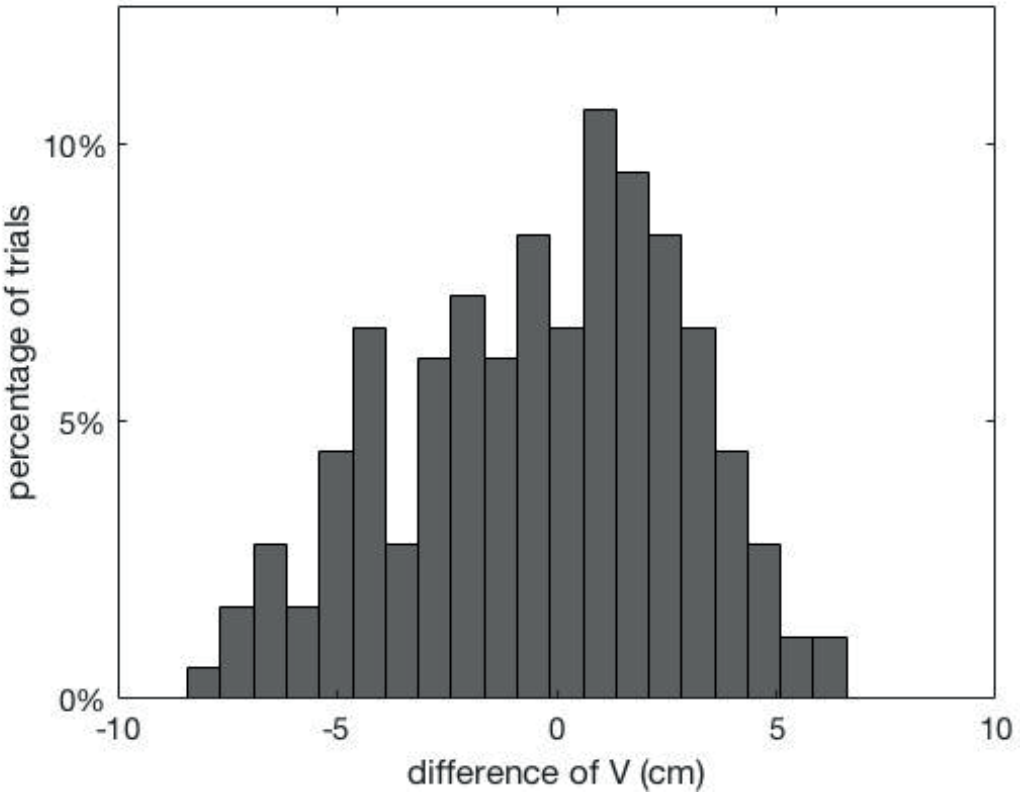


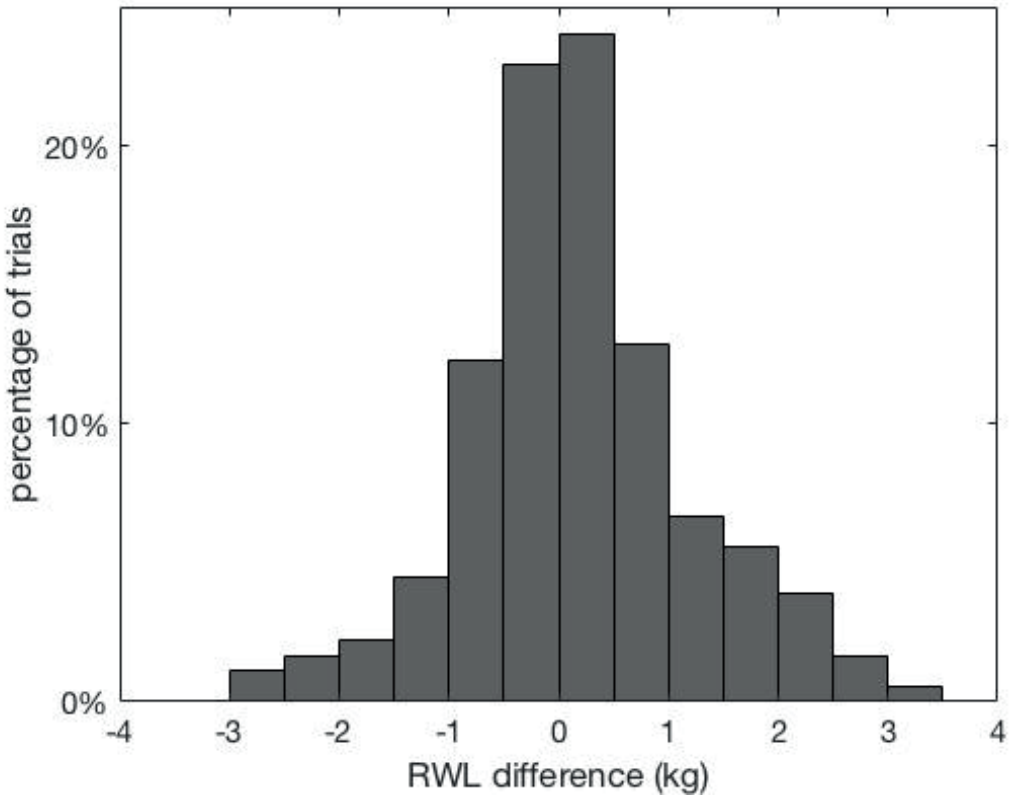
Figure 35 Histogram of the difference of V at loading

4.4.4 RWL Prediction at the Loading Instance

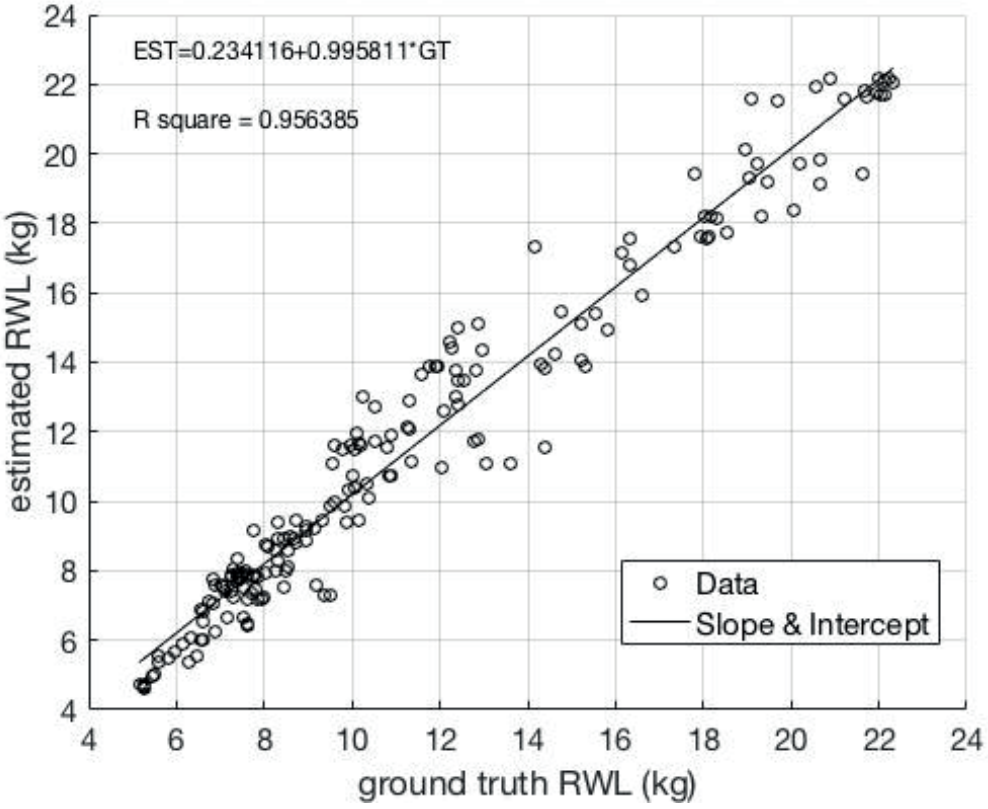
Comparing the multipliers between the computer vision estimations and motion tracking ground truth, the 95 percent confidence interval of the VM difference was 0.005 ± 0.016, the DM difference was 0.002 ± 0.011, and the HM difference was 0.013 ± 0.111. In calculating the RWL, the LC weight was 23 kg. AM was set equal to one because the subjects were asked to perform the lifting trials while keeping their torso upright without twisting. CM was also set equal to one because the effectiveness of hand-to-object coupling was considered good for all lifting trials. Since each lifting task was performed only once and was not repetitive, we calculated the Frequency Independent Recommended Weight Limit

(FIRWL) and set FM = 1. The variables V and H were the values at the loading instance. D was the vertical distance from the origin of the object to the instance before the object is loaded at the destination. VM, HM, and DM were calculated according to the multiplier transformation function required in the RNLE respectively.

The RWL difference between estimation and ground truth was calculated and shown in Figure 36 (A) where the histogram is plotted. The mean RWL difference was 0.19 kg (SD = 1.03 kg). Based on this distribution, 91 percent of the measured lifting instance detections were within ± 2 kg of the ground truth measure. The linear regression between the video estimated RWL and ground truth RWL ($R^2 = 0.96$) is shown in Figure 36 (B). A Bland-Altman analysis for the limits of agreement (mean difference ± 1.96 SD of differences) for the estimation and the motion capture data were between -1.84 kg and 2.21 kg. This demonstrates that the estimation agreed.



A



B

Figure 36 (A) Histogram of the difference of RWL at loading instance. (B) Linear regression for ground truth motion capture and video estimated RWL.

4.5 Discussion

4.5.1 Accuracy of Distances

The calculation of RWL in the validation experiment relies on the variables VM, HM, and DM, which are functions of V, H, and D, respectively. The results found the greatest bias of RWL comes

from the HM because its error range was on the order of 10^{-1}, while the error range of VM and DM was on the order of 10^{-2}. The relatively larger error in H can be explained by the following reasons:

Since the algorithm assumes that the camera viewing direction is perpendicular to the sagittal plane and thus the hands and feet should overlap because of symmetry. Therefore, the detection of the lifted object can be ascertained from either hand. However, in this validation experiment, the camera viewing angle was offset by 31.7° from the expected angle when the subjects were lifting. Consequently, the geometric center of the un-overlapped extremities would likely bias the location. Thus the validation data shows the worst-case scenario. Although the measured horizontal dimensions might not be perfectly parallel to the camera's viewing plane, the vertical dimension was always in the camera's viewing plane, so the accuracy of V was better than that of H.

4.5.2 False Negative Detections

If the algorithm cannot identify a blob satisfying the properties of a ghost blob, or the location of the detected ghost blob does not comply with human body structure, the algorithm would fail to identify a ghost blob for the video clip. Using this criterion, there were 17.13% false negative detections in this dataset.

In order to be detected by the algorithm, the ghost blob should be complete and not diluted for a consecutive sequence of N frames. To avoid the noisy blobs, an arbitrary N was set as 20 frames. If the isolated ghost blob lasted less than N frames after detaching from the subject's silhouette, the algorithm

would not identify the ghost blob. A long-duration overlap of the ghost blob and the subject's silhouette mostly depends on the subject's moving speed. In this validation dataset of six subjects, one subject moved significantly slower than the others making the blob overlap persist for a shorter time resulting in ten false negative detections for this subject, while for other subjects, there were no more than three false negative detections.

A weak appearance of an object can lead to a weaker blob. In this validation experiment, the lifted object was a thin grid, and thus its ghost blob was often weak and seen as a shadow or was diluted. Additionally, this study used a relatively low definition video (640×480 pixels) due to the limitation of the 3D motion capture system. Consequently, the plate appeared as a line segment when viewing in the plane of the plate surface. The background subtraction of this plate was represented as a 2×20 pixel image. This causes the ghost blob rapidly diminish and was considered as noise with high probability. This coarse representation undoubtedly provided less information for the algorithm, while a higher definition video should significantly improve the resolution of the blobs and thus the algorithm accuracy.

The current study was conducted in the laboratory under controlled conditions where the background was stationary and there were no extraneous motions. In an industrial setting, it is anticipated that the background might contain other workers, vehicles, or machinery that can introduce multiple ghosting images from their movements. We, therefore, use proximity to focus the ghost effect

which is caused by the subject. We plan to refine this methodology in future work in order to reject noise introduced by extraneous motions.

Since the algorithm relies on motion information to detect a moving target, if the appearance (color and tone) of the foreground and the background are close and the motion is hard to differentiate, then the algorithm is challenged and could introduce error. Since the algorithm is not sensitive to colors, the markers worn by participants for the motion capture system should not have any noticeable effect on the outcome.

Spector et al. (2014) collected a large dataset comprising six subjects performing lifting tasks in a laboratory setting. All the tasks and motions other than lifting in the trials were included for data processing. The NIOSH equation dimensions were provided using a Kinect, while the ground truth was provided by a 3D optical motion capture system. A manual inspection was performed to exclude the obvious outliers before data processing. However, in the current study, no outliers were excluded and all data was used regardless of lifting style or anomalies encountered. Comparing with the modelled data, our H estimation error was more concentrated around 0 with its 25^{th} to 75^{th} percentiles ranging within [-0.04, 0.02] m and median being -0.02 m. For the modelled data in Spector et al. (2014), the 25^{th} to 75^{th} percentiles range within about [0.06, 0.13] m, and the median is about 0.09 m (this is an average of the approximated values observed intuitively from the boxplot figures published in Spector et al. (2014)). For V estimation error, the current study results were similar to that of Spector et al. (2014) in both error

range and median. But, the current study estimation of H and V had no outliers removed, compared with the many outliers in Spector et al. (2014).

4.5.3 Recommendations for Improvements

An effective improvement in measurement error could be achieved by providing a quality assurance measurement. Some factors influencing the algorithm accuracy include, but are not limited to, how close the subject's hands and shoes are similar to the background in colour as well as whether there was proper illumination to make the significant objects clear without shadows.

A higher definition video could significantly help to detect the ghost blobs. For example, the thin plate which was represented by a 2×20 pixel blob in the current 640×480 pixel resolution video would appear as a 9×120 pixel blob in a 4K (3840×2160 pixel) resolution video. Thus it is less probable it would be eliminated as a noise blob. The videos were are limited to 640×480 pixel resolution in order to synchronize the video with the 3D motion capture data, but we anticipate that a more dense pixel resolution will provide more distinct blobs with greater contrast. Also, an object with a more obvious appearance would be helpful, e.g. substituting the current transparent thin plate into a non-transparent object. Furthermore, increased pixel resolution would provide greater precision in the distance measurements.

This validation dataset was conducted under a worst-case viewing angle (30°), and would certainly achieve a better distance estimation if the camera were set perpendicular to the sagittal plane.

The camera technology of today's devices has better resolution, the algorithms offer suitable computational simplicity, and the method has suitable stability for a hand-held camera. Additionally, an anti-shake algorithm will be explored.

In the natural work setting, the most challenging problem for this algorithm is multiple moving objects. As the final design of the video lifting monitor is implemented in a hand-held device, most of the noisy moving objects could be avoided in the image by manually choosing the recording angle.

4.6 Conclusion

In this research work, we present a 2D-video based lifting monitor. The system provides a robust, non-intrusive method to automatically extract the spatial and temporal factors necessary for applying the RNLE using a single video camera in the view from the sagittal plane. Compared with 3D motion capture, our approach provides a sufficiently high accuracy of the RWL predictions. It also provided automatic detection of lifting instances. Efficiency in computation load and non-intrusive properties will make it possible to be implemented on a hand-held device and accessible for a wide variety of applications.

Chapter 5 Body Asymmetry Angle Estimation

5.1 Abstract

Overexertion in manual lifting is one of the costliest occupational injuries and a leading cause of musculoskeletal disorders. A well-recognized and widely used risk prediction tool is the NIOSH lifting equation, which provides the recommended weight limit based on the lifting frequency, location of the object and position of the subject. In the previous chapter, we have demonstrated success automatically extracting spatial and temporal factors necessary for applying the NIOSH lifting equation using a single video camera for lifts in the view from the sagittal plane. This chapter will extend that work to varying locations throughout the workplace to process videos captured from two or more low-cost stationary cameras in arbitrary locations in the workplace to measure the lifting equation variables.

In this chapter, a body asymmetry angle estimation algorithm is proposed by estimating the 3D coordinate of each skeletal joints. We realized this by a fusion of the 2D skeletal joints estimation and the technology of structure from motion. Instead of depth cameras, 2 normal RGB cameras are used. The 2D skeletal joints estimations on each view of the camera are used as feature points for 3D reconstruction. The combination of 2D skeletal joints estimation and structure from motion contributes to the following aspects: 1. bringing structure from motion into 3D human pose estimation, 2. improving accuracy of 2D skeletal joints estimation, 3. uniting the estimation in both 2D and 3D, 4. avoiding the wearable sensors and provides more equipment flexibility in the practical application.

5.2 Body Asymmetry Angle Estimation Algorithm

5.2.1 Definition of Body Asymmetry Angle

The NIOSH has defined the asymmetry angle. It is an angular measurement of how far the object is displaced from the front (mid-sagittal plane) of the worker's body at the beginning or end of the lift, in degrees.

It is the angle between the asymmetry line and the mid-sagittal line. The asymmetry line is defined as the horizontal line that joins the mid-point between the inner ankle bones and the point projected on the floor directly below the mid-point of the hand grasps, as defined by the large middle knuckle. The sagittal line is defined as the line passing through the mid-point between the inner ankle bones and lying in the mid-sagittal plane, as defined by the neutral body position. The neutral body position is the position of the body when the hands are directly in front of the body and there is minimal twisting at the legs, torso, or shoulders.

5.2.2 NIOSH Laboratory Data

Data utilized in the current study were from a previous study conducted by researchers at the National Institute for Occupational Safety and Health (NIOSH). The original study aimed to record symmetrical lifting tasks using the combination of two lifting task variables (horizontal and vertical distances) defined by the American Conference of Governmental Industrial Hygienists (ACGIH) Threshold Limit Values (TLV) for lifting (ACGIH, 2007). The TLV for lifting classifies 12 risk zones

using the two task variables for symmetrical lifting. It is very similar to the data used in Chapter 4, except for several differences:

(1). The lifted object is a wired basket, not a wired frame.

(2). The number of valid subjects is 10, not 6. So, in total there are 360 valid video clips.

(3). The video data for each trial were recorded by a web camera (Microsoft 1080p LifeCam) and a panning camera (Sony camcorder) operated by NIOSH staff. They were both synchronized with whole-body motion data measured by a motion capture system (Optotrack 12 IR camera system, model Flex 13 with the MotionMonitor data acquisition program, Innovative Sports, Inc., Chicago, USA). The web camera was set up in a fixed location at the typical eye level and approximately 4 m in distance from the beginning of each trial. The web camera viewing angle to the subjects' initial standing point was near perpendicular to the direction of the trial. The resolution of the LifeCam video recordings was set at 640 × 480 pixels at a 30 fps rate to meet the hardware synchronization requirements for the motion capture system. The Sony camera was set almost symmetrical to the web camera about the subject's walking path. The Sony camera was fixed on a tripod with only one flexibility in horizontal yaw panning. Its panning angle was controlled by staff to follow the motion of the subject. It is recorded with resolution 1280 × 720 at 30 fps. The corresponding view of the two cameras is shown in Figure 37 and Figure 38.

Figure 37 Sony camera view

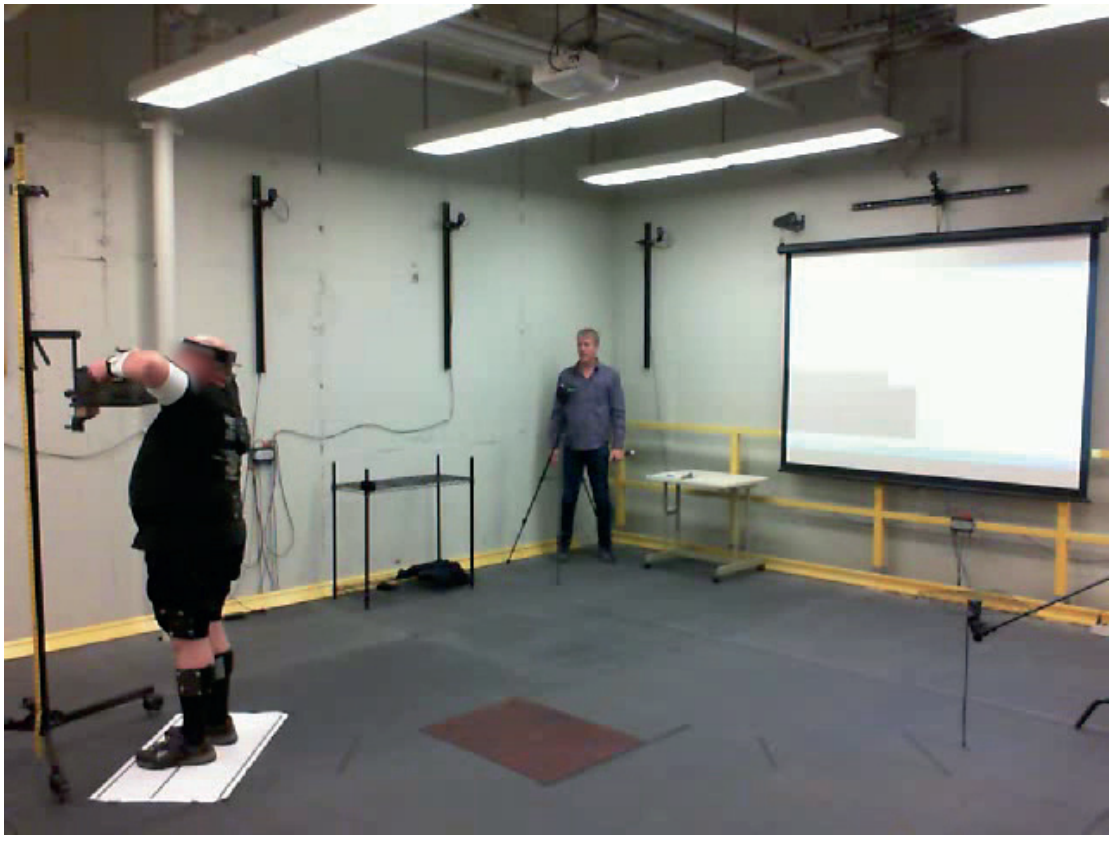


Figure 38 Life camera view

5.2.3 Overview of Computer Vision Method for Body Feature Point Location Estimation

To obtain accurate estimates of the body asymmetry angle, one needs an accurate estimate of the four body feature points, namely left and right wrist joints and ankle joints respectively.

Recent computer vision algorithms have made significant progress to estimate these body feature points from a video using deep learning techniques [95][96][97][98]. Among these, *OpenPose* [95] has shown great promise. Details of the *OpenPose* algorithm can be found in [95] and will not be discussed here. Suffice to say, the deep learning algorithm of *OpenPose* using thousands of examples of human body images can produce an estimate of an underlying skeleton model of body torso and limbs. While individual joint locations may not be as accurate as we want, *OpenPose* does offer a coarse estimate of the desirable body feature points. However, these initial estimates of the four body feature points have been found less accurate for use of body asymmetry angle estimation.

To remedy this problem, in this work, we leverage the availability of video streams from the dual camera set up in the NIOSH's experiment to obtain two sets of body feature points estimates. Then we developed a novel computer vision algorithm to iteratively refine the location estimates of these body points to improve the accuracy of the body asymmetry angle estimates. The tasks performed by our algorithm can be summarized as follows:

a. Camera calibration: taking advantage of some shared stationary feature points of the scene, use structure from motion (SfM) [99] to estimate the intrinsic and extrinsic parameters of each camera as well as the 3D coordinates of these stationary feature points.

b. Feature point localization: Use the estimated essential matrix and epipolar constraints imposed by the pin-hole camera model to estimate the 3D coordinates of the joint locations of the pair of 2D body skeleton models estimated from the two cameras using *OpenPose*.

c. 3D Location Refinement of the feature point: Iteratively perturb the 3D location estimates of the body feature points (ie. Wrists, ankles) so that the epipolar constraints will be better satisfied while the constraints of a 3D skeleton model are also satisfied.

The detailed derivation of these computer vision algorithms will be discussed in the next section.

5.3 Computer Vision Algorithm Development

5.3.1 Camera Calibration and Lifting Frame Determination

The inputs to the computer vision algorithm are video streams from two video cameras shooting the worker performing the lifting operation. We assume these two cameras are properly synchronized in time so that video frames corresponding to the same time instant can be identified. Based on these videos from both cameras, we present two approaches for estimating the body asymmetry angle during a lifting operation. Before applying these algorithms, we will calibrate the intrinsic parameters of each

camera. This is done by identifying a set of corresponding feature points in a pair of video frames with the same time index. With at least 18 pairs of corresponding feature points, two 3 × 3 intrinsic parameter matrices $\mathbf{K}_i$, i = 1, 2 may be estimated together with the six extrinsic parameters specifying the relative positions of the pair of cameras at that time instant. In our work, the camera calibration is performed using structure from motion functions from Matlab [105]. In the videos data used in this work, one camera is rotated to follow the worker walking toward the lifting station. Hence, although the camera calibration algorithm can also calibrate the extrinsic parameters [**R** | **t**] (**R** contains the three rotation angles; **t** is the translation vector), these parameters may need to be re-estimated for the lifting frame.

The next step is to detect the time stamp of the lifting operation. This will be accomplished using a motion-based algorithm proposed earlier [100]. The video analytic algorithm then will be performed only for video frames at the lifting instant.

5.3.2 Structure from Motion (SfM) Using Estimated 2D Joint Positions

Structure from Motion (SfM) [99] is a computer vision technique to jointly calibrate a pair of cameras while estimating the 3D positions of a set of feature points jointly shared by both views. Several different formulations and corresponding implementations are available as open-source programs [102][103][104][105]. In this work, we use the SfM tools from Matlab [105]. The steps are briefly described as follows:

1) Feature Extraction

Two types of 2D feature points will be extracted: background feature points and body feature points. A Harris feature extractor [106] is used to detect a set of distinct 2D feature points from each frame. The outcome contains a set of 2D coordinates and corresponding feature descriptor vectors. A RANSAC [101] feature matching algorithm will use these feature descriptors and corresponding 2D positions to establish the correspondence of detected 2D feature points from each view. These background feature points will be used to establish the relative camera poses (extrinsic parameters) at the lifting frame.

We also extract a set of 2D body feature points using an open-source package *OpenPose* [107]. Given a video frame containing the body image of the worker performing a lifting task, *OpenPose* can estimate up to 25 2D joints (key points) positions of the worker. These 2D joint positions are indexed with the pre-defined joint positions and hence correspondence from both cameras are readily available.

2) Relative Camera Pose Estimation

The epipolar constraint will be used to estimate the relative camera pose of one camera versus the other. Denote $\tilde{x}$ and $\tilde{\mathrm{x}}'$ respectively to be 3D homogeneous image coordinates at a pair of pin-hole cameras corresponding to the same 3D points. The relative pose of camera #2 with respect to camera #1 (reference) can be specified by a rotation matrix (**R**) and a translation vector (**t**). Define an *essential matrix*

$$\mathbf{E} = [\mathbf{t}]_{\times}\mathbf{R}, \tag{5.1}$$

where $[\mathbf{t}]_{\times}$ is a 3×3 *cross-product* matrix defined by elements of the vector **t**. The epipolar constraint

then stipulates that

$$\tilde{x}^T \boldsymbol{E} \tilde{x}' = 0 \tag{5.2}$$

Given 8 or more pairs of background feature points $\tilde{x}$ and $\tilde{x}'$, the essential matrix **E** may be estimated using singular value decomposition [108]. Then the camera pose **R** and **t** can be derived accordingly.

3) Estimating 3D Body Joint Coordinate Using Triangulation

Given a pair of calibrated cameras (known $\mathbf{K}_i$, i = 1, 2, **R**, and **t**), we estimate the 3D coordinates of left and right wrists and ankles from their 2D pixel positions estimated using *OpenPose*. Denoting **p** to be the 4 × 1 homogeneous coordinate of a joint point. The relation between **p** and the homogeneous image coordinate $\tilde{x}$ is described by

$$\tilde{x} = \alpha \begin{bmatrix} x_k \\ y_k \\ 1 \end{bmatrix} = \mathbf{K} \cdot [\mathbf{R}\ \mathbf{t}]\mathbf{p} = \mathbf{M}\mathbf{p} = \begin{bmatrix} \mathbf{m}_1^T \mathbf{p} \\ \mathbf{m}_2^T \mathbf{p} \\ \mathbf{m}_3^T \mathbf{p} \end{bmatrix} \tag{5.3}$$

where **M** is a 4 × 3 matrix and $\mathbf{m}_i^T$, i = 1, 2, 3 are the three rows of the **M** matrix. (x_k, y_k) is the 2D coordinate of the k^{th} feature point corresponding to **p** and α is a constant relating to the depth of **p**. Hence one may write $x_k = (\mathbf{m}_1{}^T\mathbf{p})/(\mathbf{m}_3{}^T\mathbf{p})$ and $y_k = (\mathbf{m}_2{}^T\mathbf{p})/(\mathbf{m}_3{}^T\mathbf{p})$. In other words,

$$\begin{cases} \left(\mathbf{m}_1^T \mathbf{p}\right) - x_k \cdot \left(\mathbf{m}_3^T \mathbf{p}\right) = 0 \\ \left(\mathbf{m}_2^T \mathbf{p}\right) - y_k \cdot \left(\mathbf{m}_3^T \mathbf{p}\right) = 0 \end{cases}$$

Or in matrix form $$\underbrace{\begin{bmatrix} 1 & 0 & -x_k \\ 0 & 1 & -y_k \end{bmatrix}}_{\mathbf{H}_k} \begin{bmatrix} \mathbf{m}_1^T \\ \mathbf{m}_2^T \\ \mathbf{m}_3^T \end{bmatrix} \mathbf{p} = \mathbf{H}_k \mathbf{M} \mathbf{p} = \mathbf{0} \tag{5.4}$$

Note that both $\mathbf{H}_k$ and $\mathbf{M}$ are known. Let $\mathbf{G}_k$ be a matrix of the same format as $\mathbf{H}_k$ formed by the pixel coordinates of the feature point (x_k', y_k') at the other camera corresponding to $\mathbf{p}$. Then, one has

$$\begin{bmatrix} \mathbf{H}_k \\ \mathbf{G}_k \end{bmatrix}_{4\times 3} \mathbf{M}_{3\times 4} \mathbf{p}_{4\times 1} = \mathbf{M}'_{4\times 4} \mathbf{p}_{4\times 1} = \mathbf{0} \tag{5.5}$$

Note that the matrix $\mathbf{M}$' is rank-deficient. One may solve $\mathbf{p}$ as the right singular vector corresponding to the minimum singular value of the matrix $\mathbf{M}$'. This process may be repeated to estimate the 3D coordinates of each of the four desired joint positions.

5.3.3 Body Asymmetry Angle Estimation

In the experiment, the subjects are instructed to perform this activity with both left and right body parts symmetrically. Thus we use the line segment between the pair of wrists to represent the upper body direction, and use the line segment connecting the pair of ankles to represent the lower body direction. The asymmetry angle would be the intersection angle between these two directions.

Let P_{LW}, P_{RW}, P_{LA}, and P_{RA} be x and y components of the estimated 3D coordinates of the left-, right-side wrists and the left-, right- ankles of the worker during lifting. Each of them is a 2×1 vector. One may compute a *wrist* direction vector as:

$$\mathbf{v} = P_{RW} - P_{LW} = [v_1 \; v_2]^T \tag{5.6}$$

Then the wrist angle $\theta_W = \tan^{-1}(v_2/v_1)$. Similarly, one may compute the ankle orientation angle θ_A. Finally, the body asymmetry angle θ_{BT} may be estimated as:

$$\theta_{BT} = \theta_W - \theta_A. \tag{5.7}$$

5.3.4 Accuracy Assessment

The accuracy of the estimation of θ depends on the accuracy of the 3D body feature points. Taking derivatives of the θ_W formula, one has (the subscript W is omitted for brevity):

$$\frac{d\tan\theta}{d\theta} = \frac{1}{v_1^2}\left(v_1\frac{dv_2}{d\theta} - v_2\frac{dv_1}{d\theta}\right) = \frac{1}{\cos^2\theta} = 1+\tan^2\theta \quad -\pi/2 < \theta < \pi/2$$

Or $$d\theta = \frac{d\tan\theta}{1+\tan^2\theta} = \frac{(v_1 dv_2 - v_2 dv_1)/v_1^2}{1+(v_2/v_1)^2} = \frac{v_1 dv_2 - v_2 dv_1}{v_1^2+v_2^2} = \frac{\begin{bmatrix} v_1 & -v_2 \end{bmatrix}}{\|\mathbf{v}\|^2} d\mathbf{v} = \begin{bmatrix}\cos\theta & -\sin\theta\end{bmatrix}\frac{d\mathbf{v}}{|\mathbf{v}|} \tag{5.8}$$

Above equation provides a way to gauge the estimation error $d\theta$ in terms of the estimation error of the joint positions $\mathbf{v}$ and joint position estimation error $d\mathbf{v}$. In the above equation, $\cos\theta = v_1/|\mathbf{v}|$, and $\sin\theta = v_2/|\mathbf{v}|$. Note that the term $d\mathbf{v}/|\mathbf{v}|$ is a relative error which increases as $|\mathbf{v}|$ reduces. Here $|\mathbf{v}|$ is the 3D horizontal distance between the wrists or between the ankles.

The estimation error of body feature points (wrists, ankles) may be estimated empirically. One useful indicator for a specific feature point is the minimum singular value of the corresponding **M'** matrix as defined in eq. (5.5). The estimate of a feature point would be less accurate if it is associated with a larger minimum singular value. Another metric would be the quadratic epi-polar constraint in eq. (5.2). An example is shown in Figure 42. The epipolar constraint for the skeleton joints has been

calculated. It is expected that if the feature points pairs from both of the camera views are having correct locations, then their corresponding epipolar constraint should be 0. However, for the *OpenPose* result on each camera views, the epipolar constraint result deviates from 0. Comparatively, the skeleton joints after triangulation adjustment become closer to 0. The epipolar constraint for the static points is closer to 0. It is because the static points are easier to be pointed out by the algorithm with a specific and accurate location.

To explore the relationship between the angle and the ending point of a uni-vector, we take the uni-vector of hip at lifting instant for exploration. For x coordinate of hip uni-vector, the mean value is 0.001430, and the standard deviation is 0.003714. For y coordinate of hip uni-vector, the mean value is -0.002819, and the standard deviation is 0.001982. The distribution of the hip uni-vector ending point is shown in Figure 39.

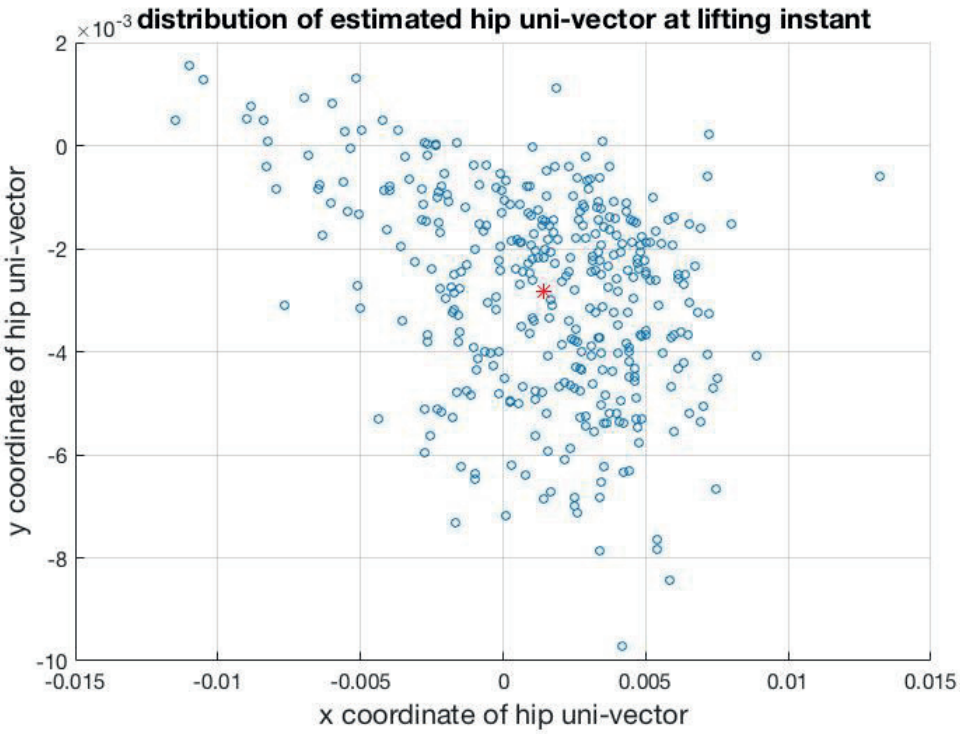


Figure 39 Distribution of the hip uni-vector ending point

Based on the statistical values of hip uni-vector, we plot the change of hip uni-vector angle when the x coordinate of hip uni-vector changes from (mean-std) to (mean+std). As shown in Figure 40, the angle varies a lot.

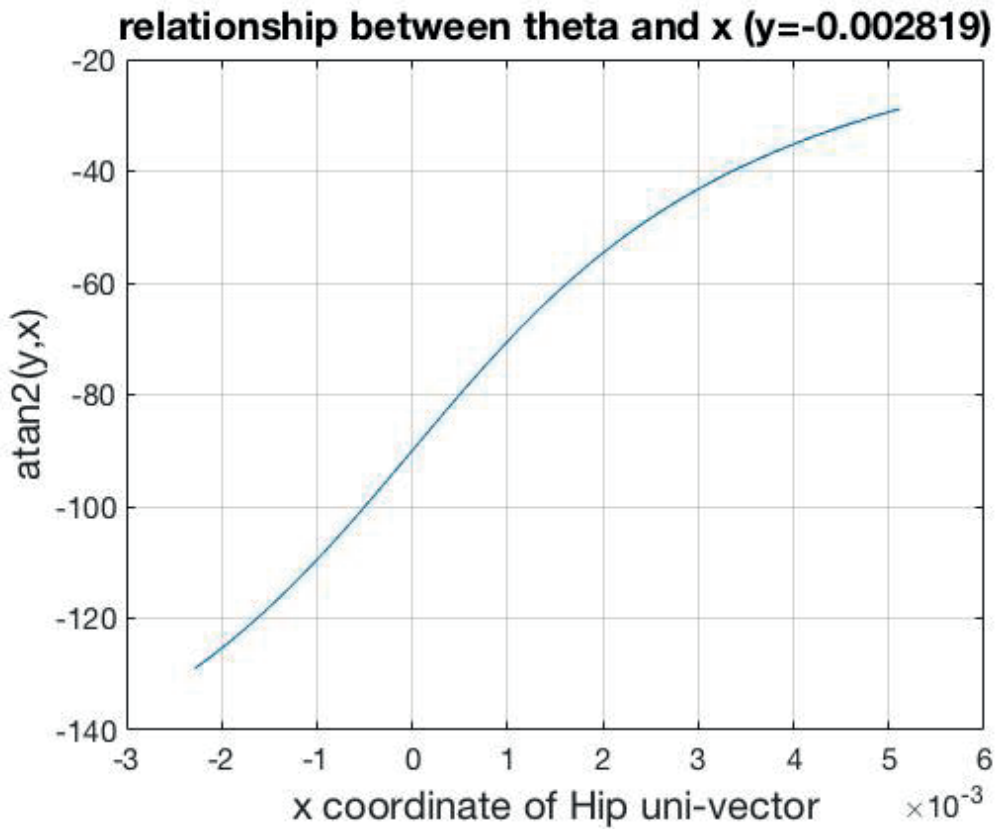


Figure 40 Relationship between asymmetry angle and x coordinate of hip uni-vector ending point

We plot the change of hip uni-vector angle when the y coordinate of hip uni-vector changes from (mean-std) to (mean+std). As shown in Figure 41, the angle varies a lot. These figures show that the accuracy of 3D skeleton joints estimation affects the accuracy of asymmetry angle very much.

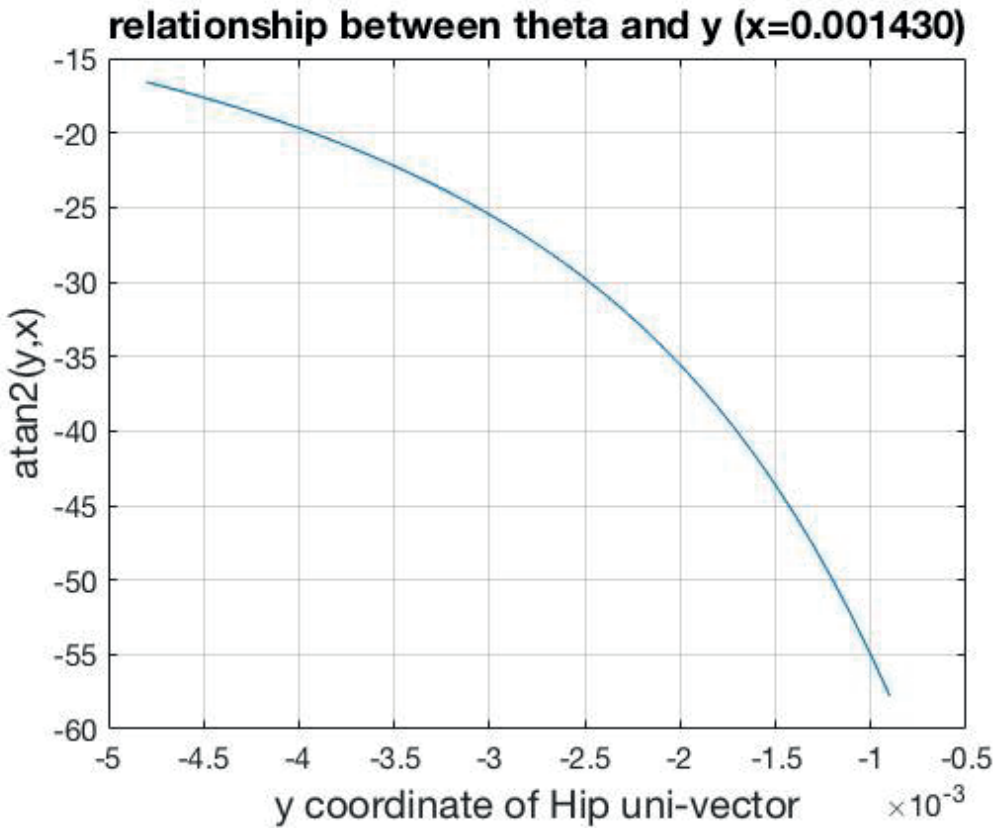


Figure 41 Relationship between asymmetry angle and y coordinate of hip uni-vector ending point

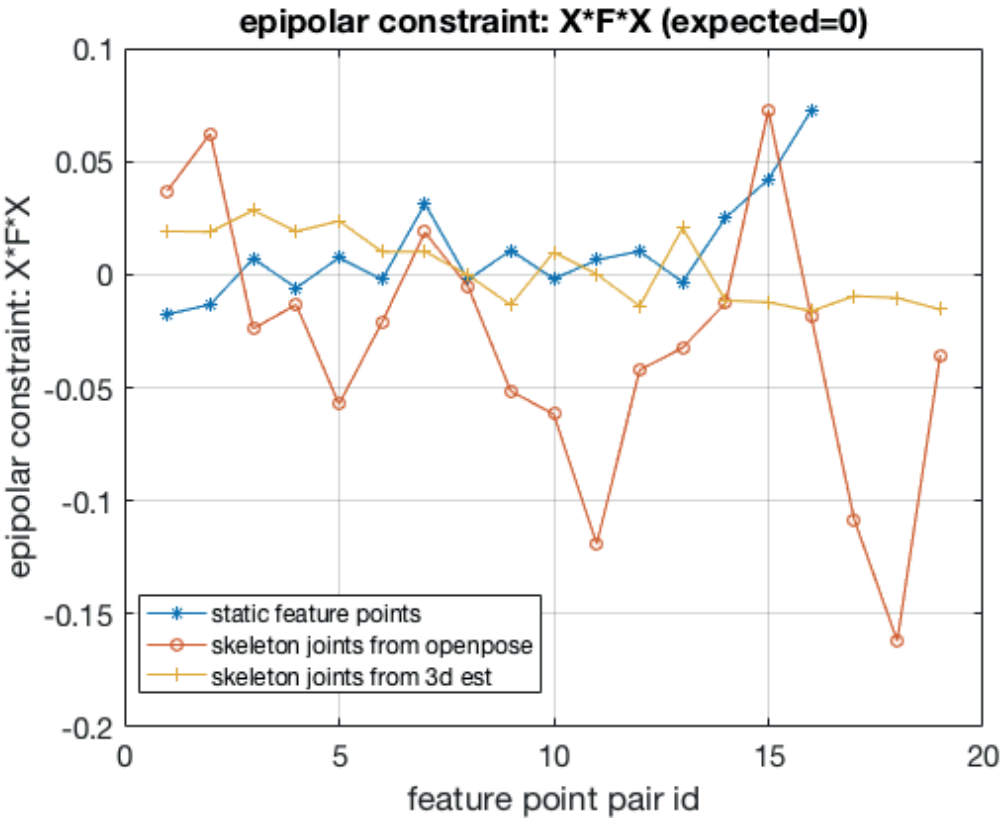


Figure 42 Epipolar constrain on different pairs of feature points

5.4 Experiment

5.4.1 Asymmetry Angle Estimation

The experiment is carried out on the *NIOSH* laboratory dataset. The asymmetry angle is calculated following the formula (5.6) - (5.7). The output angle is in degrees. The estimated angle is calculated using the estimated 3D skeletal joints coordinates, and the MoCap angle is calculated using the 3D skeletal joints coordinates from the MoCap sensors. The difference between the estimated angle and the

MoCap angle is calculated for the lifting instant of all the 360 video clips.

At the lifting instant, the asymmetry angle should be around 0°. The mean of the difference angle is 3.9°, and the standard deviation is 16.5°. A histogram of the difference angle is plotted in Figure 43. It has 81.4% confidence that the estimated asymmetry angle is within (-15°, 15°) difference from the MoCap asymmetry angle.

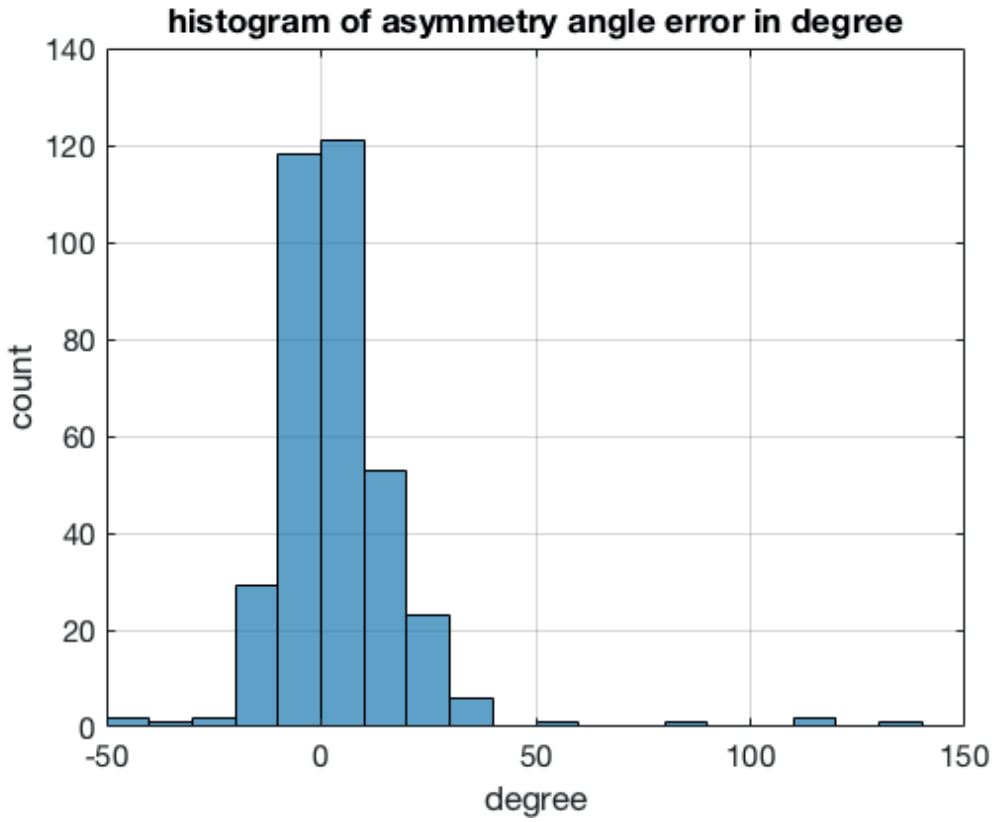


Figure 43 Histogram of asymmetry angle difference

Figure 44 shows the distribution of estimated angle and MoCap based angle. The difference between these two distributions is in Figure 45. The mean of the difference angle is 0.4°, and the standard deviation is 21.4°. It has 35.3% confidence that the estimated asymmetry angle is within (-7.5°, 7.5°) difference

from the MoCap asymmetry angle.

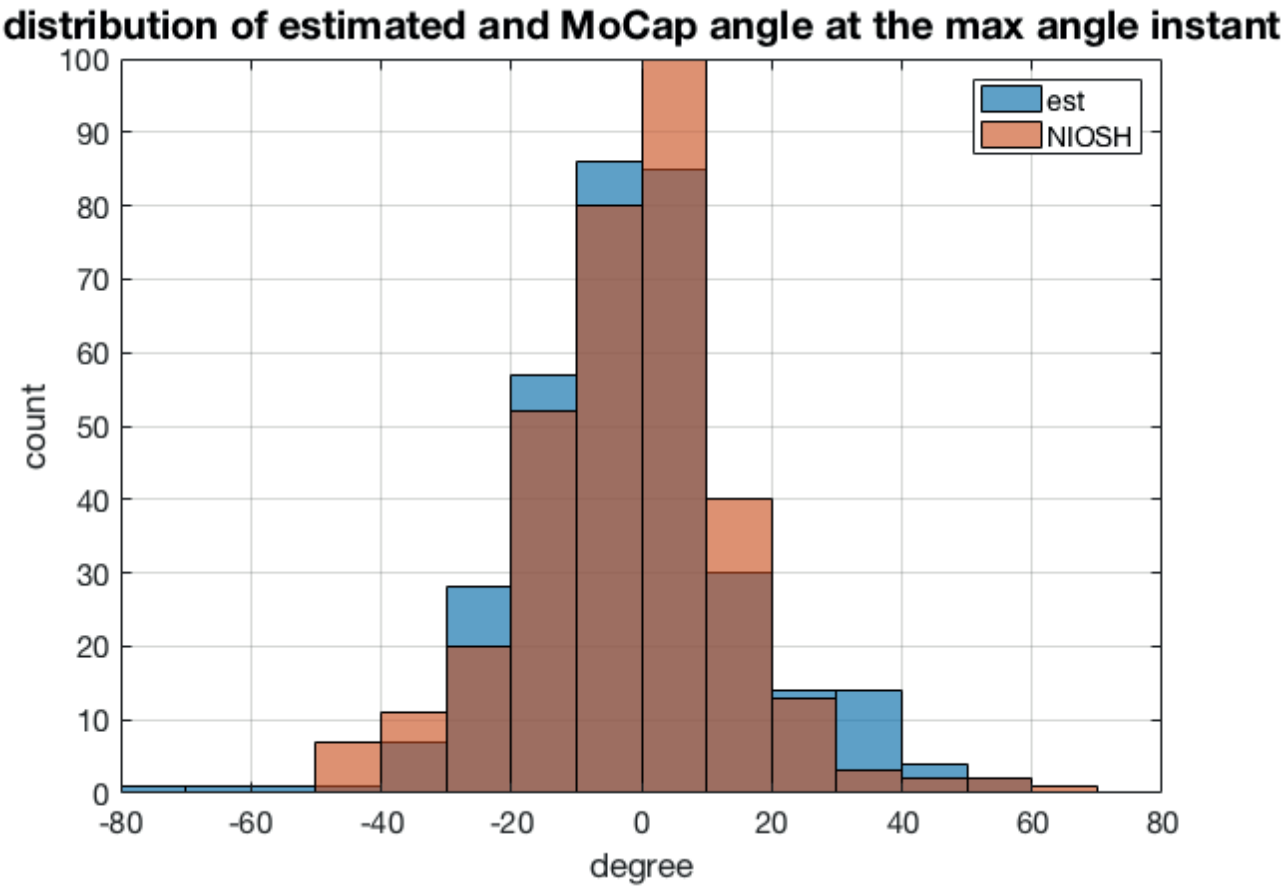


Figure 44 Histogram of estimated and MoCap based asymmetry angle

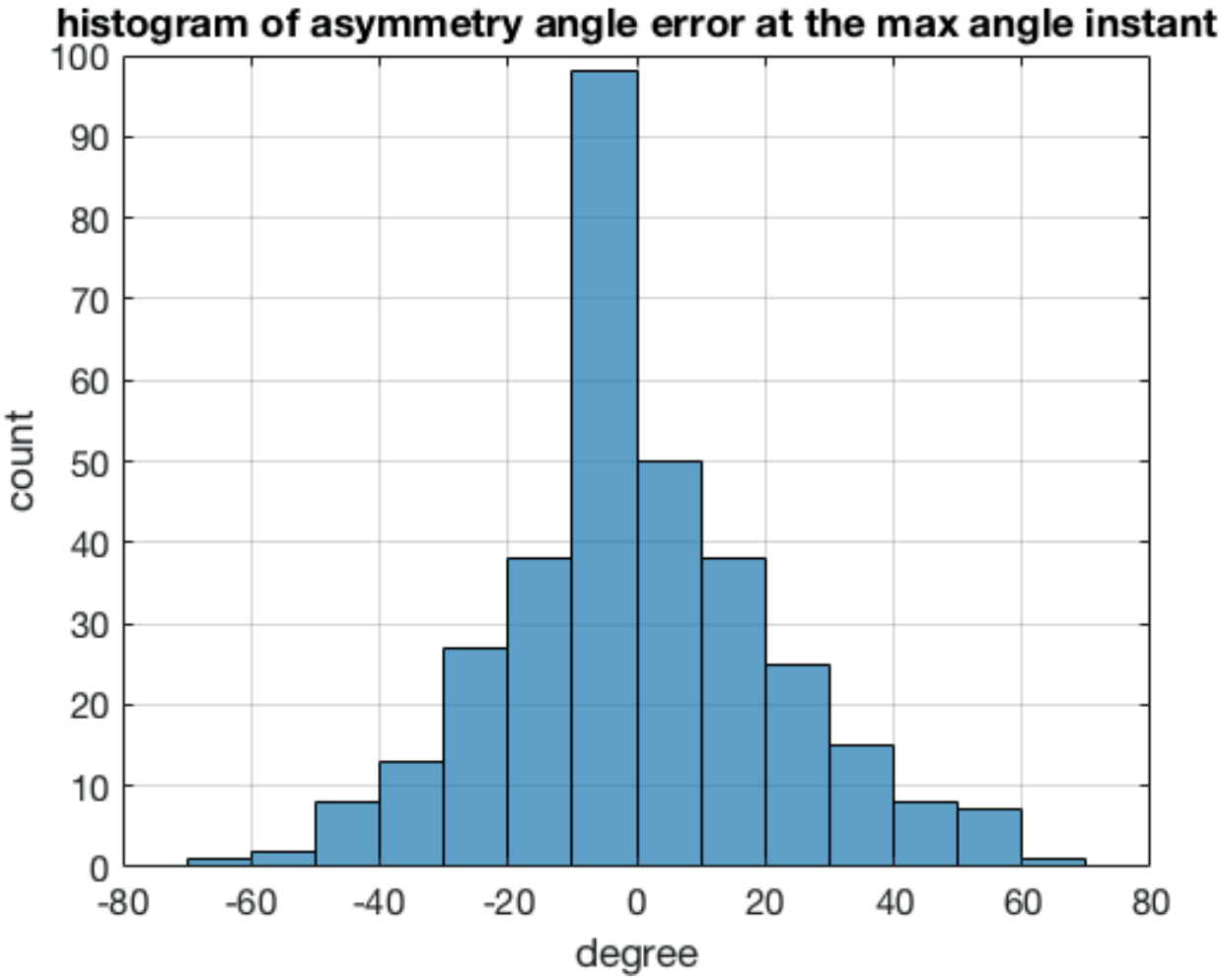


Figure 45 Histogram of angle difference

5.4.2 Outliers Analysis

Though the MoCap system is not guaranteed correct, we'd like to refer to it as the ground truth. The experimental result shows the estimated asymmetry angle was approximately close to the ground truth angle, with the center of the error distribution centered around 0°. However, the 3-sigma range was about -48° to 48°, while the error tolerance was designed as 15° according to the asymmetry angle

category range in the NIOSH lifting equation element converting table. An example of an acceptable estimated angle is shown in Figure 46.

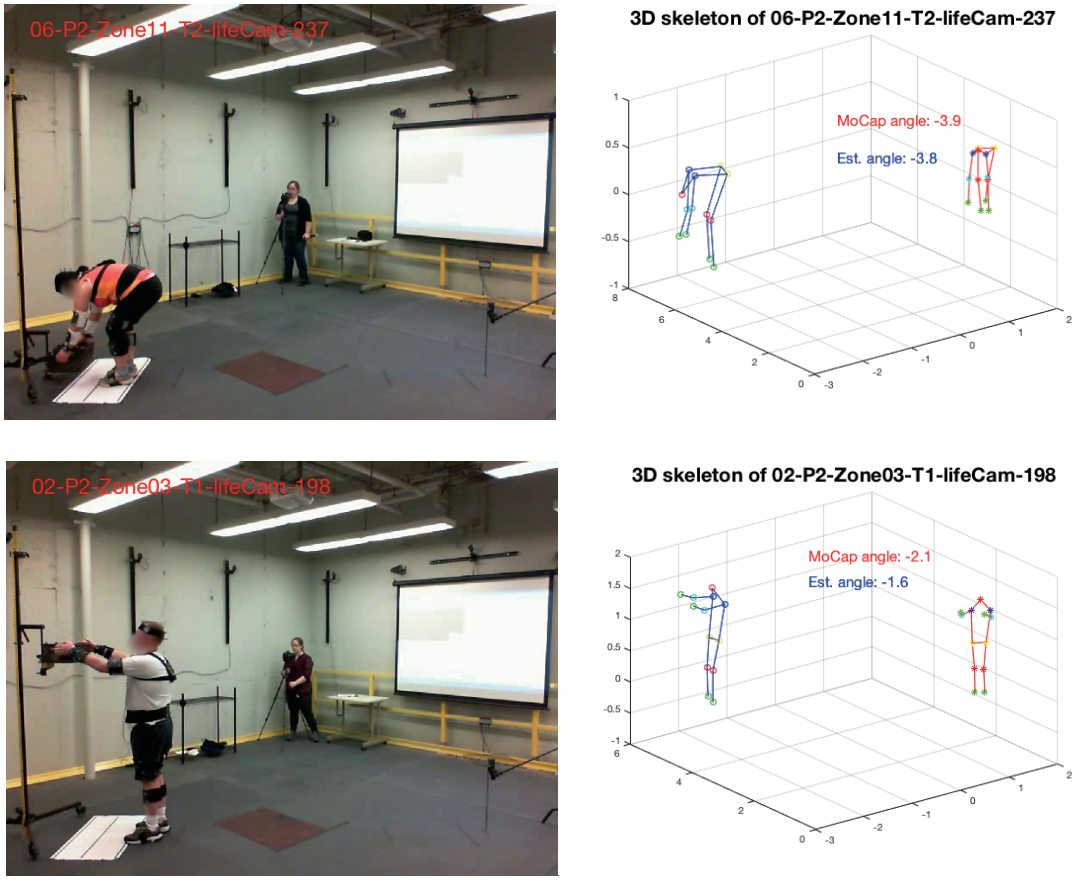
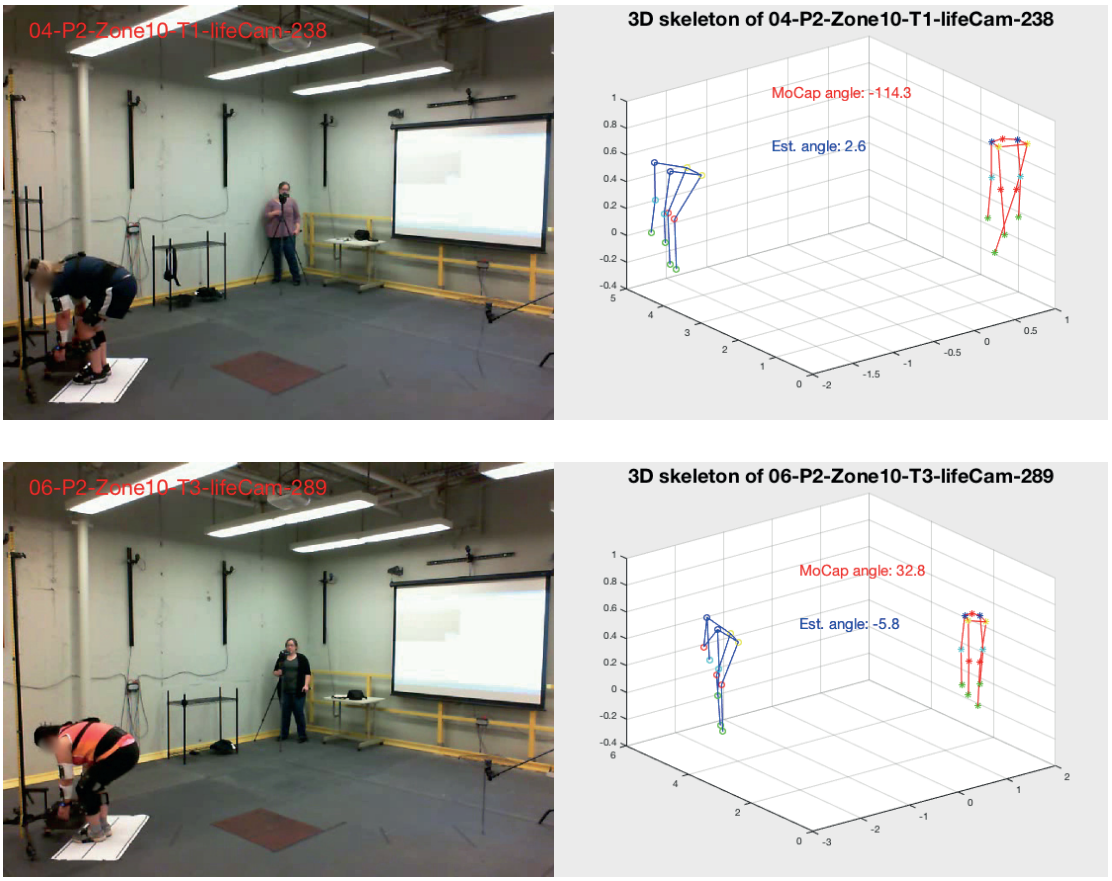


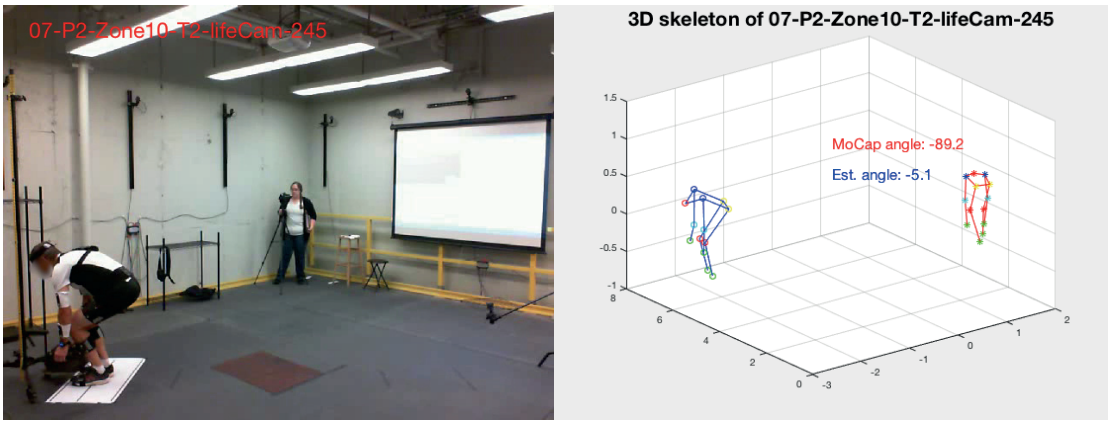
Figure 46 The examples of acceptable estimation. The left column is the video frame where the lifting instant takes place, and the corresponding video name and the frame number is labeled in red at the upper left corner. The right column is the corresponding 3D skeleton joints from MoCap sensor (red skeleton) and the estimation (blue skeleton). The corresponding asymmetry angle is listed in the middle of the plot. The head is plotted in red, the shoulders are plotted in blue, the elbows are plotted in cyan, the wrists are plotted in green, the hips are plotted in yellow, the knees are plotted in red, and the ankles are plotted in green. In the first row, the estimated asymmetry angle is -3.8°, and it is very close to the MoCap asymmetry angle

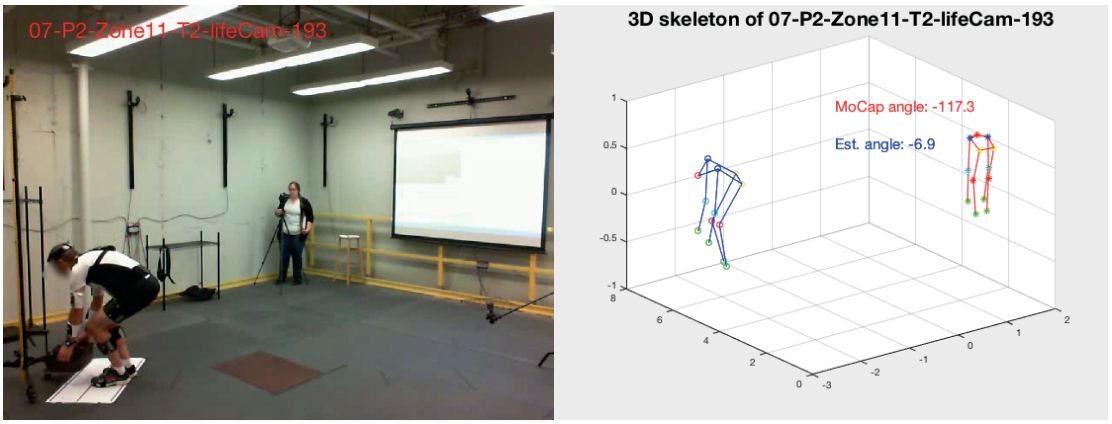
-3.9°. And in the second row, the estimated asymmetry angle is -1.6°, and it is very close to the MoCap asymmetry angle -2.1°.

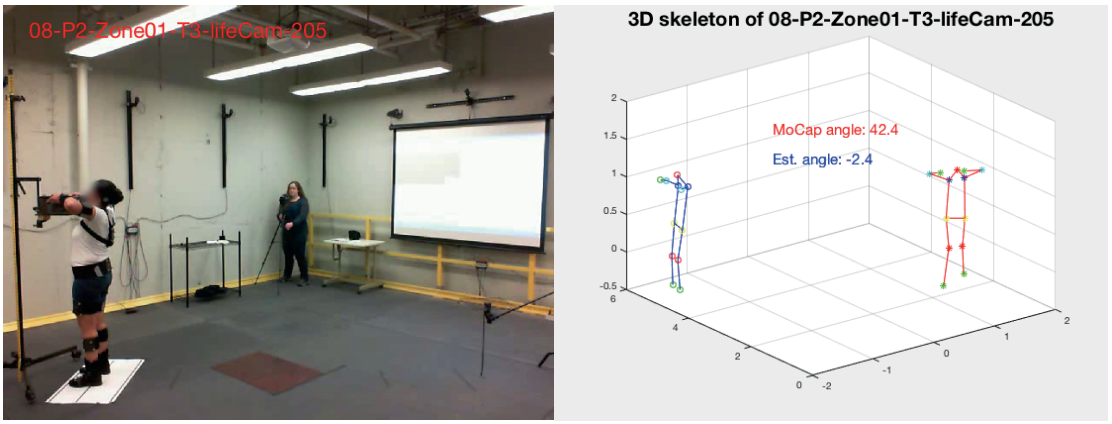
We look at the outliers where the difference angle is larger than 40° or smaller than -20°. There are 10 cases in total, and we summarize the reasons for these outliers as following:

(1) The MoCap sensor data is wrong. For example, in the first row of Figure 47, the left and right wrist from the MoCap data is twisted. This doesn't coincide with the video image, and it gives a big asymmetry angle. There are 7 cases among these 10 outliers that are caused by this problem. The estimated asymmetry angle coincides with the video image, thus the difference angle is large.




07-P2-Zone10-T2-lifeCam-245
3D skeleton of 07-P2-Zone10-T2-lifeCam-245
MoCap angle: -89.2
Est. angle: -5.1


07-P2-Zone11-T2-lifeCam-193
3D skeleton of 07-P2-Zone11-T2-lifeCam-193
MoCap angle: -117.3
Est. angle: -6.9


08-P2-Zone01-T3-lifeCam-205
3D skeleton of 08-P2-Zone01-T3-lifeCam-205
MoCap angle: 42.4
Est. angle: -2.4

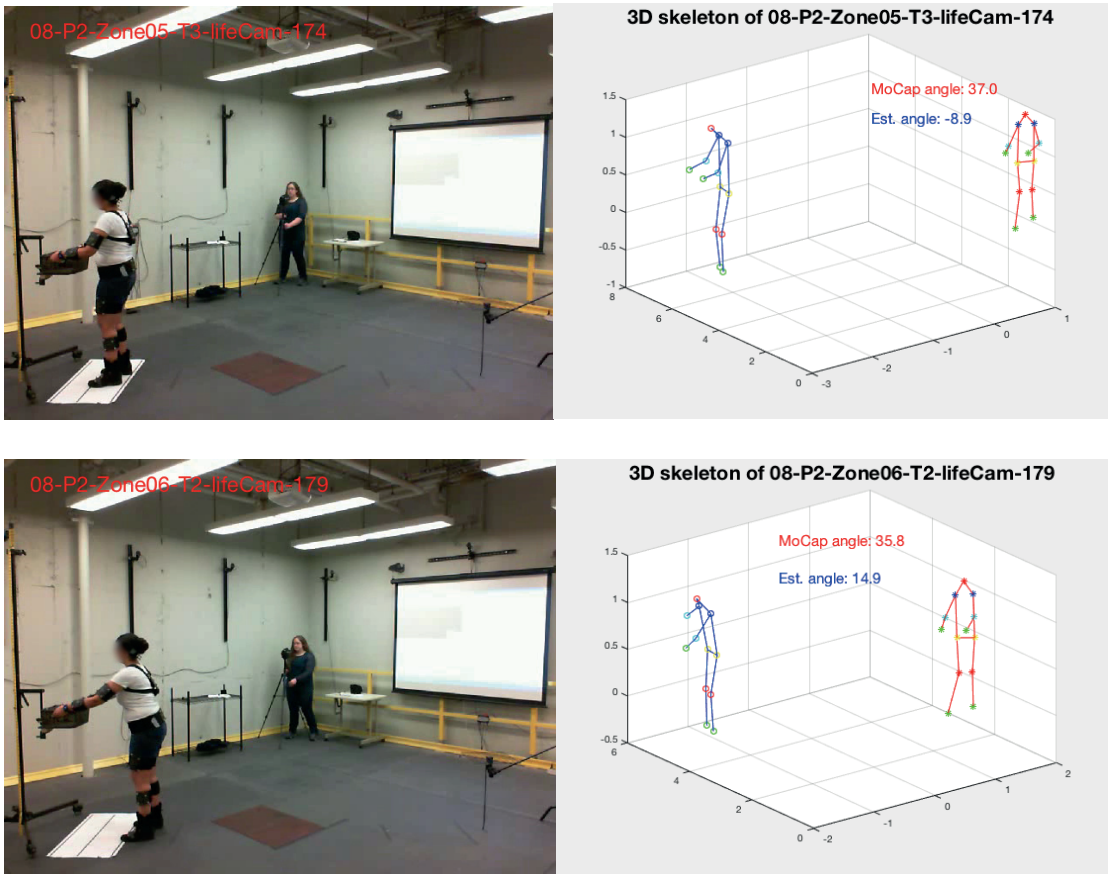
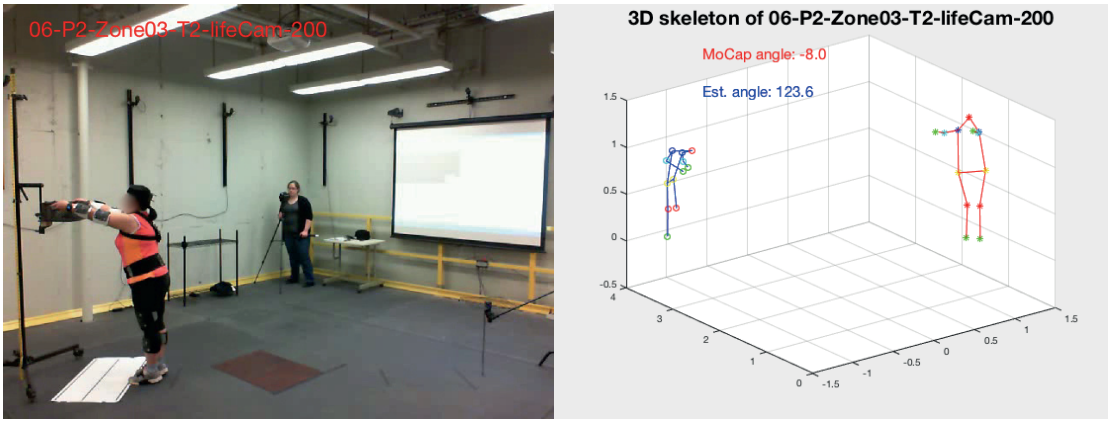


Figure 47 Examples of the erroneous MoCap data cases.

(2) The estimated 3D skeleton joints coordinates are erroneous. In this situation, the estimated skeleton is twisted, and its corresponding asymmetry angle is biased from the MoCap angle.



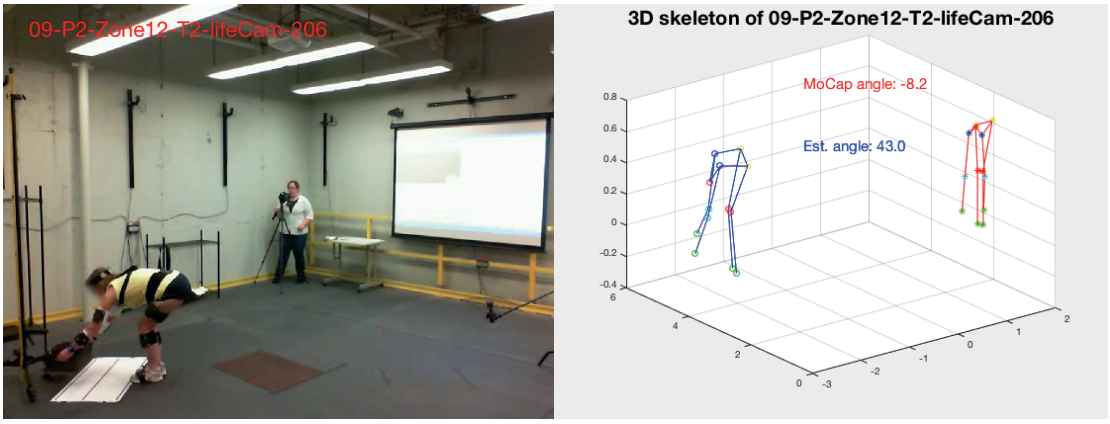


Figure 48 Examples of the erroneous estimated cases.

(3) MoCap angle and the estimated angle are not in the same direction. Due to the bias of 3D skeleton joints location, the wrist direction vector and the ankle direction vector are slightly tilted. However, if the titled direction is not consistent for the MoCap data and the estimated data, e.g. Mocap data showing the upper body is twisted towards right while the estimated angle showing left, the difference angle will be very large.

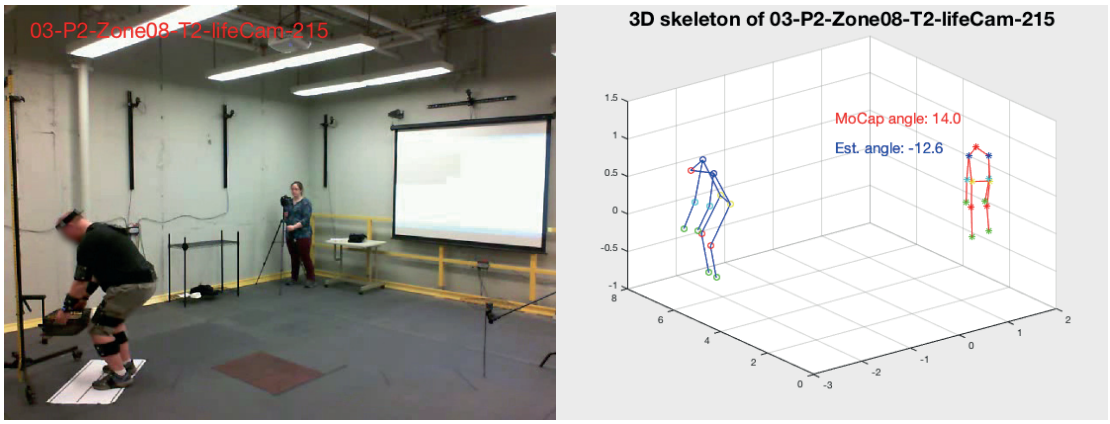


Figure 49 The example of the different twisting direction.

5.4.3 Asymmetry Angle Difference Analysis

After eliminating these 10 outliers, the mean value becomes 3.1°, and the standard deviation becomes 10.7°. Still, there is quite some difference between the estimated asymmetry angle and the MoCap based asymmetry angle. Several factors are contributing to this difference:

(a). Several body parts were missing from the 3D skeletal joint estimation. Since the technique used here was structure from motion, any point could get its 3D coordinate estimated once it is seen by both of the cameras. However, in this lab setup, each camera is set almost at the subject's sagittal plane, that means half of the body could not be seen because of occlusion. And also, since the cameras are set in the opposite direction, a lot of body parts can't be viewed together by both of the cameras. So, in this dataset, a lot of left or right wrist, elbow or ankle couldn't get its 3D coordinate estimation because it is not viewed simultaneously by both of the cameras. In the situation of missing wrist, the algorithm would take both of the elbows to calculate the wrist direction vector **v** in formula (6). If unfortunately the elbow is missing as well, both of the shoulders would be used. Both of the knees would be used if any ankle is missing.

(b). Erroneous feature points extraction and matching. In this structure from the motion algorithm, the extrinsic parameters of the cameras are calculated based on the matching feature points pairs showing up in both of the camera views. The accuracy of each of the feature point location affects the accuracy of the estimation of the extrinsic parameters very much. In this algorithm, the feature points are extracted by the algorithm, and the feature points are not with sharp corners or clear edges due to

shadow and video resolution (shown in Figure 50). Thus, the location of each feature point could be biased from frame to frame using this feature extractor. And this leads to a biased camera calibration, which further leads to a biased 3D skeletal joints estimation.

(a) (b)

Figure 50 Crops of video frames. It shows the feature points don't have a clear corner to be marked out.

(c). Skeletal joints definition was not consistent between *OpenPose* and the 3D motion capture system. The 3D skeletal joints coordinate estimation was based on the output of a 2D skeletal joint coordinate estimator *OpenPose*. In *OpenPose*, each body part is represented by a point according to how each body part is labeled by a point in the training dataset. While in Motion capture system, each body part is represented by a point as well according to the skeletal model it used. However, it could be possible that *OpenPose* and the motion capture system were not using the same point to represent a body part. In addition to these factors, the accuracy of *OpenPose* output, and the accuracy of Motion capture system also affect the asymmetry angle estimation performance.

5.5 Conclusion

In this research, an algorithm to estimate the asymmetry angle for the revised NIOSH lifting equation (RNLE) was developed. This asymmetry angle estimation algorithm combines structure from motion and CNN based 2D skeletal model estimator, provides an advanced structure from motion

algorithm to estimate 3D skeletal joints coordinates. It provides an additional approach to make an accurate and detailed 3D body pose estimation.

The experiment shows the estimated angle is close to the Motion Capture based angle. The inaccuracy caused by occlusion and camera calibration is to be improved in the future. However, the quality of the Motion Capture data also requires validation. A further manual quality check might have to be carried out to provide reliable ground truth.

To summarize, this algorithm together with this team's previous work of estimating spatial and temporal factors using motion-based information, could be used to automatically calculate the RNLE. The experiments conducted show that this approach provides a high-accuracy estimation, making a hand-held automatic and convenient lifting risk predictor realizable.

Chapter 6

6.1 Summary

In this thesis, we focused on developing efficient methods for video-based human activity analysis with specific applications for monitoring driver distraction and ergonomics of factory workers performing heavy lifting jobs.

In the application of monitoring driver distraction, the goal was to reduce the computation time of the VOT algorithm without significantly compromising the overall tracking performance. We proposed a novel temporal frame-subsampling scheme that will process a shorter sub-sampled video sequence and use interpolation to estimate object locations. The theoretical analysis and experiments show that this approach could greatly reduce computation time while not sacrificing accuracy.

In the application of lifting monitoring, the goal was to reduce the complexity of the VOT algorithm when applied to track factory workers working in an industrial environment, performing heavy lifting jobs. We proposed an efficient and practical approach to automatically extract spatial and temporal factors necessary for applying the RNLE. It leverages motion information to detect the lifting instant, segment the foreground from background, and detect the location of hands and feet. In the follow-up work, it predicts 3D skeletal model by taking the 2D skeletal joints estimation as input feature points for structure from motion. The asymmetry angle is then calculated based on this 3D skeletal model. The experiment shows the proposed method is very promising and making a convenient and hand-held

lifting monitoring device realizable. And one software has got started to be built up utilizing this algorithm to step towards the realization of a hand-held lifting monitoring device.

6.2 Future Research Directions

We conclude this thesis by suggesting some future research directions.

- **Improving tracking performance by utilizing the similarity function from One-shot learning.** Although we have discussed a method to detect drifting, it is still necessary to improve the tracking accuracy to avoid drifting from the source. One-shot learning, as a deep neural network trained from a huge number of images, it is providing an effective similarity function that provides a reliable similarity score when only one template is given.

- **Explore a few more state-of-art 2D human skeletal joints estimators.** The current approach uses *OpenPose* as a 2D skeletal joints feature extractor. However, its accuracy is limited and it couldn't handle the situation when the partial body is occluded. This affects the accuracy of the 3D skeletal joints estimation. Exploring more 2D skeletal joints estimator would improve the accuracy of our asymmetry angle estimator.

Acknowledgements

This material is based upon work supported by the US Dept of Transportation, Federal Highway Administration under contract number DTFH6114C00011, the grants from the National Institute for Occupational Safety and Health (NIOSH/CDC), R01OH011024 (Radwin), and the Grant or Cooperative Agreement Number, T42 OH008455, funded by the Centers for Disease Control and Prevention. Its contents are solely the responsibility of the authors and do not necessarily represent the official views of the Centers for Disease Control and Prevention or the Department of Health and Human Services.

References

[1]. Yilmaz, A., Javed, O., and Shah, M., "Object tracking: A survey." *Acm computing surveys (CSUR)* 38.4 (2006): 13

[2]. Wang, X., Hu, Y. H., Radwin, R. G., and Lee, J. D., "Head tracking using video analytics." in *IEEE Global Conf on Signal and Information Processing (GlobalSIP),* 2015.

[3]. Kalal, Z., Mikolajczyk, K., and Matas, J., "Tracking-learning-detection." *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 7 (2012): 1409-1422.

[4]. Babenko, B., Yang, M. H., and Belongie, S., "Robust object tracking with online multiple instance learning." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.8 (2011): 1619-1632.

[5]. Hare, S., Golodetz, S., Saffari, A., Vineet, V., Cheng, M. M., Hicks, S. L., and Torr, P. H., "Struck: Structured output tracking with kernels." *IEEE transactions on pattern analysis and machine intelligence* 38.10 (2016): 2096-2109

[6]. Grabner, H., Leistner, C., and Bischof, H., "Semi-supervised on-line boosting for robust tracking."*Computer Vision–ECCV 2008* (2008): 234-247

[7]. Zhong, W., Lu, H., and Yang, M. H., "Robust object tracking via sparsity-based collaborative model." *Computer vision and pattern recognition (CVPR), 2012 IEEE Conference on. IEEE*, 2012.

[8]. Wu, Y., Lim, J., and Yang, M. H., "Online object tracking: A benchmark." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2013.

[9]. Kristan, M., *et al*. "The visual object tracking vot2013 challenge results." *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2013.

[10]. LIRIS, F., "The Visual Object Tracking VOT2014 challenge results."

[11]. Kristan, M., *et al*. "The visual object tracking vot2015 challenge results." *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2015.

[12]. Kristan, M., *et al*. "The Visual Object Tracking VOT2016 Challenge Results. " In: Hua G., Jégou H. (eds) Computer Vision – ECCV 2016 Workshops. *ECCV 2016. Lecture Notes in Computer Science,* vol 9914.

[13]. Kristan, M., *et al*. "The Visual Object Tracking VOT2017 Challenge Results" *The IEEE International Conference on Computer Vision (ICCV)*2017.

[14]. Campbell, K. L., "The SHRP 2 naturalistic driving study: Addressing driver performance and behavior in traffic safety." *TR News* 282 (2012).

[15]. Long-Term Detection and Tracking, *CVPR* 2014 http://www.micc.unifi.it/LTDT2014/

[16]. Wang, X., Hu, Y. H., Radwin, R. G., and Lee, J. D., “Frame-Subsampled, Drift-Resilient Video Object Tracking”, *2018 International Conference on Acoustics, Speech, and Signal Processing* (IEEE ICASSP 2018)

[17]. Wang, X., Hu, Y. H., Radwin, R. G., and Lee, J. D., “Frame-Subsampled, Drift-Resilient Long-Term Video Object Tracking”, *IEEE International Conference on Multimedia and Expo (ICME)* 2018

[18]. Wang, X., Hu, Y. H., Radwin, R. G., and Lee, J. D., "Temporal Frame Sub-Sampling for Video Object Tracking", *Journal of Signal Processing Systems* (2019): 1-13.

[19]. Wang, X., Hu, Y. H., Lu, M. L., Radwin, R. G., " The Accuracy of a 2D Video-Based Lifting Monitor", *Ergonomics* just-accepted (2019): 1-33

[20]. Henriques, J. F., Caseiro, R., Martins, P., and Batista, J., "High-speed tracking with kernelized correlation filters." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.3 (2015): 583-596.

[21]. Kılıç, V., Barnard, M., Wang, W., and Kittler, J., "Audio assisted robust visual tracking with adaptive particle filtering." *IEEE Transactions on Multimedia*17.2 (2015): 186-200.

[22]. De Freitas, A., Mihaylova, L., Gning, A., Angelova, D., and Kadirkamanathan, V., "Autonomous crowds tracking with box particle filtering and convolution particle filtering." *Automatica*69 (2016): 380-394.

[23]. Klein, J., Peters, C., Martin, J., Laurenzis, M., and Hullin, M. B., "Tracking objects outside the line of sight using 2D intensity images." *Scientific reports* 6 (2016): 32491.

[24]. Ochs, P., Malik, J., and Brox, T., "Segmentation of moving objects by long term video analysis." *IEEE transactions on pattern analysis and machine intelligence* 36, no. 6 (2014): 1187-1200.

[25]. He, J., Balzano, L., and Szlam. A., "Incremental gradient on the grassmannian for online foreground and background separation in subsampled video." *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE,* 2012.

[26]. Leung, A. P., and Gong. S., "Optimizing distribution-based matching by random subsampling." *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on. IEEE,* 2007.

[27]. Multi-resolution video analysis and key feature preserving video reduction strategy for (real-time) vehicle tracking and speed enforcement systems, by Wu, W., Bernal, E.A., Loce, R.P., and Hoover, M.E. (2015, Feb 10) U.S. Patent 8,953,044.

[28]. Nam, H. and Han, B., "Learning multi-domain convolutional neural networks for visual tracking." *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*. IEEE, 2016.

[29]. Held, D., Thrun, S., and Savarese, S., "Learning to track at 100 fps with deep regression networks." *European Conference on Computer Vision*, pp. 749-765. Springer, Cham, 2016.

[30]. Korshunov, P., and Ooi. W. T., "Reducing frame rate for object tracking." *International Conference on Multimedia Modeling*. Springer, Berlin, Heidelberg, 2010.

[31]. Misra, I., Shrivastava, A., and Hebert, M., "Watch and learn: Semi-supervised learning of object detectors from videos." In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference* on, pp. 3593-3602. IEEE, 2015.

[32]. Parseval des Chênes, Marc-Antoine Mémoire sur les séries et sur l'intégration complète d'une équation aux différences partielles linéaire du second ordre, à coefficients constants" presented before the Académie des Sciences (Paris) on 5 April 1799. This article was published in *Mémoires présentés à l'Institut des Sciences, Lettres et Arts, par divers savants, et lus dans ses assemblées. Sciences, mathématiques et physiques.* (Savants étrangers.), vol. 1, pages 638–648 (1806).

[33]. Supancic, J. S., and Ramanan, D., "Self-paced learning for long-term tracking." *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2013.

[34]. Ma, C., et al. "Long-term correlation tracking." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015.

[35]. Chen, C.-H., Hu, Y. H., Yen, T. Y., and Radwin, R. G., "Automated Video Exposure Assessment of Repetitive Hand Motion," *Human Factors,* 55(2), pp. 298 – 308, 2013.

[36]. Viola, P., and Jones, M., "Rapid object detection using a boosted cascade of simple features." *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference* on.Vol. 1. IEEE, 2001

[37]. Sivaraman, S. and Trivedi, M. M., "Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1773-1795, Dec. 2013.

[38]. Sivaraman, S. and Trivedi, M. M., "A General Active-Learning Framework for On-Road Vehicle Recognition and Tracking," in *IEEE Transactions on Intelligent Transportation Systems,* vol. 11, no. 2, pp. 267-276, June 2010.

[39]. Dalal, N., and Triggs, B., “Histograms of oriented gradients for human detection,” in *Proc. IEEE Comput. Soc. Conf. CVPR,* 2005, pp. 886–893.

[40]. Freund, Y., and Schapire, R., “A short introduction to boosting,” *J. Jpn. Soc. Artif. Intell.*, vol. 14, no. 5, pp. 771–780, Sep. 1999.

[41]. Cortes, C. and Vapnik, V., “Support vector networks,” *Mach. Learn.,* vol. 20, no. 3, pp. 273–297, Sep. 1995.

[42]. Mukhtar, A., Xia, L. K., and Tang, T. B., "Vehicle detection techniques for collision avoidance systems: A review." *IEEE Transactions on Intelligent Transportation Systems* 16.5 (2015): 2318-2338.

[43]. Krause, J., et al. "3d object representations for fine-grained categorization." *Proceedings of the IEEE International Conference on Computer Vision Workshops.* 2013.

[44]. http://web.mit.edu/torralba/www/database.html

[45]. Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J., & Black, M. J. (2016, October). Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *European Conference on Computer Vision* (pp. 561-578). Springer, Cham.

[46]. Bureau of Labor Statistics (2016): https://www.bls.gov/news.release/osh2.nr0.htm

[47]. Delpresto, J., Duan, C., Layiktez, L. M., Moju-Igbene, E. G., Wood, M. B., & Beling, P. A. (2013, April). Safe lifting: an adaptive training system for factory workers using the Microsoft Kinect. *In IEEE Systems and Information Engineering Design Symposium, SIEDS* (pp. 64-69).

[48]. Dempsey PG (2002). Usability of the revised NIOSH lifting equation. *Ergonomics* 45:817–828.

[49]. Gabel, M., Gilad-Bachrach, R., Renshaw, E., and Schuster, A. (2012, August). Full body gait analysis with Kinect. *In Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE* (pp. 1964-1967). IEEE.

[50]. Howe, N. R., Leventon, M. E., and Freeman, W. T. (2000). Bayesian reconstruction of 3d human motion from single-camera video. In *Advances in neural information processing systems* (pp. 820-826).

[51]. Juul-Kristensen, B., Hansson, G. Å., Fallentin, N., Andersen, J. H., and Ekdahl, C. (2001). Assessment of work postures and movements using a video-based observation method and direct technical measurements. *Applied ergonomics*, 32(5), 517-524.

[52]. Kim, S., and Nussbaum, M. A. (2013). Performance evaluation of a wearable inertial motion capture system for capturing physical exposures during manual material handling tasks. *Ergonomics*, 56(2), 314-326.

[53]. Li, G., and Buckle, P. (1999). Current techniques for assessing physical exposure to work-related musculoskeletal risks, with emphasis on posture-based methods. *Ergonomics*, 42(5), 674-695.

[54]. Liberty Mutual Research Institute for Safety. Liberty Mutual Workplace Safety Index, Liberty Mutual 175 Berkeley St., Boston, MA 02116 (2017).

[55]. Lowe, B., Dempsey, P., Jones, E., and National Institute for Occupational Safety and Health (NIOSH). (2018, September). Assessment Methods Used by Certified Ergonomics Professionals. *In Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 62, No. 1, pp. 838-842). Sage CA: Los Angeles, CA: SAGE Publications.

[56]. Patrizi, A., Pennestrì, E., and Valentini, P. P. (2016). Comparison between low-cost marker-less and high-end marker-based motion capture systems for the computer-aided assessment of working ergonomics. *Ergonomics*, 59(1), 155-162.

[57]. Plantard, P., Shum, H. P., Le Pierres, A. S., and Multon, F. (2017). Validation of an ergonomic assessment method using Kinect data in real workplace conditions. *Applied ergonomics*, 65, 562-569.

[58]. Meherizi, R., Xu, X., Zhang, S., Pavlovic, V., Metaxas, D., and Li, K. (2017). Using a marker-less method for estimating L5/S1 moments during symmetrical lifting. *Applied Ergonomics.* 65, 541-550

[59]. Meherizi, R., Peng, X., Xu, X., Zhang, S., and Li, K. (2018a). A computer vision based method for 3D posture estimation of symmetrical lifting. *Journal of Biomechanics*. 69, 40-46.

[60]. Meherizi, R., Peng, X., Xu, X., Zhang, S., Metaxas, D., and Li, K. (2018b). Estimating 3D L5/S1 joint load during lifting using a deep learning based method. *IEEE Transactions on Human-Machine Systems, One-line.*

[61]. Spector, J. T., Lieblich, M., Bao, S., McQuade, K., & Hughes, M. (2014). Automation of workplace lifting hazard assessment for musculoskeletal injury prevention. *Annals of occupational and environmental medicine*, 26(1), 15.

[62]. Waters, T. R., Putz-Anderson, V., Garg, A., and Fine, L. J. (1993). Revised NIOSH equation for the design and evaluation of manual lifting tasks. *Ergonomics*, 36(7), 749-776.

[63]. Zhao, W., Xu, L., Bai, J., Ji, M., & Runge, T. (2018). Sensor-based risk perception ability network design for drivers in snow and ice environmental freeway: a deep learning and rough sets approach. *Soft computing*, 22(5), 1457-1466.

[64]. Zhao, W., Xu, L., Xi, S., Wang, J., & Runge, T. (2017). A Sensor-Based Visual Effect Evaluation of Chevron Alignment Signs' Colors on Drivers through the Curves in Snow and Ice Environment. *Journal of Sensors*, 2017.

[65]. Zhao, W., Xu, L., Dong, Z. S., Qi, B., & Qin, L. (2018). Improving transfer feasibility for older travelers inside high-speed train station. *Transportation Research Part A: Policy and Practice,* 113, 302-317.

[66]. Hao, Y., Xu, L., Qi, B., Wang, T., & Zhao, W. (2019). A Machine Learning Approach for Highway Intersection Risk Caused by Harmful Lane-Changing Behaviors. In *CICTP 2019*(pp. 5623-5635).

[67]. Zhao, W., Yin, J., Wang, X., Hu, J., Qi, B., & Runge, T. (2019). Real-time vehicle motion detection and motion altering for connected vehicle: algorithm design and practical applications. *Sensors*, 19(19), 4108.

[68]. Qi, B., Liu, P., Ji, T., Zhao, W. and Banerjee, S., "DrivAid: Augmenting Driving Analytics with Multi-Modal Information," *2018 IEEE Vehicular Networking Conference (VNC),* Taipei, Taiwan, 2018, pp. 1-8.

[69]. Zivkovic, Z. (2004, August). Improved adaptive Gaussian mixture model for background subtraction. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on* (Vol. 2, pp. 28-31). IEEE.

[70]. Zivkovic, Z., & Van Der Heijden, F. (2006). Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern recognition letters,* 27(7), 773-780.

[71]. Wang, X, Hu, Y. H., Radwin, R. G., and Lee, J. D., "Head tracking using video analytics." *Signal and Information Processing (GlobalSIP), 2015 IEEE Global Conference on. IEEE, 2015.*

[72]. US Department of Commerce (2013), The Geographic Concentration of Manufacturing Across the United States. Retrieved from http://www.esa.doc.gov/sites/default/files/finalthegeographicconcentrationofmanufacturingacrosstheunitedstates.pdf

[73]. Ji, T., et al. "Fast and efficient integration of human upper-body detection and orientation estimation in RGB-D video." *Communication Software and Networks (ICCSN), 2017 IEEE 9th International Conference on. IEEE, 2017*.

[74]. Rybok, L., et al. "Multi-view based estimation of human upper-body orientation." *Pattern Recognition (ICPR), 2010 20th International Conference on. IEEE,* 2010.

[75]. Raza, M., et al. "Appearance-based pedestrians' head pose and body orientation estimation using deep learning." *Neurocomputing* 272 (2018): 647-659.

[76]. Smith, B. M., et al. "Automatic Driver Face State Estimation in Challenging Naturalistic Driving Videos 2." *Transportation Research Board 95th Annual Meeting*, Washington, DC. 2016.

[77]. Shoushtarian, B., and Helmut E. B. "A practical adaptive approach for dynamic background subtraction using an invariant color model and object tracking." *Pattern Recognition Letters* 26.1 (2005): 5-26.

[78]. Seitz, S. M., and Dyer, C. R., "View morphing." *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. *ACM*, 1996.

[79]. Cheung, K. M., Baker, S., and Kanade, T., "Shape-from-silhouette across time part i: Theory and algorithms." *International Journal of Computer Vision* 62.3 (2005): 221-247.

[80]. Wu, Y., Lim, J., and Yang, M. H., "Online object tracking: A benchmark." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2013.

[81]. Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pugfelder, R., et al.: The sixth Visual Object Tracking VOT2018 challenge results. In: *ECCV2018 Workshops, Workshop on visual object tracking challenge* (2018)

[82]. Li, F. F., Fergus, R., and Perona, P., "One-shot learning of object categories." *IEEE transactions on pattern analysis and machine intelligence* 28.4 (2006): 594-611.

[83]. Li, B., et al. "High Performance Visual Tracking With Siamese Region Proposal Network." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2018.

[84]. Valmadre, J., et al. "End-to-end representation learning for correlation filter based tracking." *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on. IEEE, 2017.*

[85]. Huang, C., Lucey, S., and Ramanan, D., "Learning policies for adaptive tracking with deep feature cascades." *IEEE Int. Conf. on Computer Vision (ICCV)*. 2017.

[86]. Matthews, I., Ishikawa, T., and Baker, S., "The template update problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 810–815, Jun. 2004.

[87]. Stalder, S., Grabner, H., and Gool, L. V., "Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition," *in Proc. 12th IEEE Int. Conf. Comput. Vis. Workshop*, 2009, pp. 1409–1416.

[88]. Grabner, H., Leistner, C., and Bischof, H., "Semi-supervised on-line boosting for robust tracking," in *Proc. 10th Eur. Conf. Comput. Vis.*, 2008, pp. 234–247.

[89]. Mei, X., Ling, H., Wu, Y., Blasch, E., and Bai, L., "Minimum error bounded efficient L1 tracker with occlusion detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.,* 2011, pp. 1257–1264.

[90]. Bertinetto, L., et al. "Fully-convolutional siamese networks for object tracking." *European conference on computer vision*. Springer, Cham, 2016.

[91]. Russakovsky, O., et al. "Imagenet large scale visual recognition challenge." *International Journal of Computer Vision* 115.3 (2015): 211-252.

[92]. Bolme, D. S., et al. "Visual object tracking using adaptive correlation filters." *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010*.

[93]. Vojir, T., Noskova, J., and Matas, J., "Robust scale-adaptive mean-shift for tracking." *Pattern Recognition Letters49* (2014): 250-258.

[94]. Radwin, R. G., Wang, X., Hu, Y. H. and Difranco, N., "Movement Monitoring System", P170280US02

[95]. Cao, Z, et al. "Realtime multi-person 2d pose estimation using part affinity fields." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017.

[96]. Fang, H. S., et al. "Rmpe: Regional multi-person pose estimation." *Proceedings of the IEEE International Conference on Computer Vision*. 2017.

[97]. Insafutdinov, E., et al. "Arttrack: Articulated multi-person tracking in the wild." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017.

[98]. Toshev, A., and Szegedy, C., "Deeppose: Human pose estimation via deep neural networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.

[99]. Ullman, S., "The interpretation of structure from motion." Proceedings of the Royal Society of London. Series B. *Biological Sciences* 203.1153 (1979): 405-426.

[100]. Wang, X., Hu, Y. H., Lu, M. L., and Radwin, R. G., "The Accuracy of a 2D Video-Based Lifting Monitor." *Ergonomics* just-accepted (2019): 1-33.

[101]. Fischler, M. A., and Bolles, R. C., "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography." *Communications of the ACM* 24.6 (1981): 381-395.

[102]. https://github.com/mapillary/OpenSfM

[103]. https://github.com/colmap/colmap

[104]. https://github.com/NLeSC/structure-from-motion

[105]. https://www.mathworks.com/help/vision/examples/structure-from-motion-from-multiple-views.html

[106]. Harris, C. G., and Stephens, M., "A combined corner and edge detector." In *Alvey vision conference*, vol. 15, no. 50, pp. 10-5244. 1988.

[107]. https://github.com/CMU-Perceptual-Computing-Lab/openpose, Cao2017CVPR

[108]. Hartley, R. I. (June 1997). "In Defense of the Eight-Point Algorithm". *IEEE Transactions on Pattern Recognition and Machine Intelligence*. 19 (6): 580–593. doi:10.1109/34.601246.

[109]. Newell, A., Yang, K., and Deng, J., “Stacked hourglass networks for human pose estimation” *CoRR*, Vol. abs/1603.06937.2016

[110]. Yu, F., and Koltun, V., "Multi-scale context aggregation by dilated convolutions", in *ICLR*, 2016

[111]. Ionescu, C., Papava, D., Olaru, V. and Sminchisescu, C., Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, No. 7, July 2014

[112]. Greene, R.L., Hu, Y.H., Difranco, N., Wang, X., Lu., M.L., Bao,S., Lin, J.H., and Radwin, R.G. (2018, August). Predicting Sagittal Plane Lifting Postures from Image Bounding Box Dimensions. *20th Congress of International Ergonomics Association*. Florence, Italy.

[113]. Greene, R.L., Hu, Y. H., Difranco, N., Wang, X., Lu, M. L., Bao, S., Lin, J.-H .& Radwin, R. G. (2018, August). Automated Video Lifting Posture Classification Using Bounding Box Dimensions. In *Congress of the International Ergonomics Association* (pp. 550-552). Springer, Cham.

[114]. Greene, R. L., Hu, Y. H., Difranco, N., Wang, X., Lu, M. L., Bao, S. Lin, J.-H. & Radwin, R. G. (2019). Predicting Sagittal Plane Lifting Postures From Image Bounding Box Dimensions. *Human factors*, 61(1), 64-77.