3D Front-View Human Upper Body Pose Estimation Using Single Camera

Ruizhi Sun

A thesis submitted in partial fulfillment of the
requirements for the degree of:

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

University of Washington

2013

Committee:

Jenq-Neng Hwang

Linda Shapiro

Program Authorized to Offer Degree:

Department of Electrical Engineering

University of Washington

Abstract

3D Front-View Human Upper Body Pose Estimation Using Single Camera

Ruizhi Sun

Chairperson of the Supervisory Committee:
Professor: Jenq-Neng Hwang
Department of Electrical Engineering

3D human pose estimation is an important field in Computer Vision. It has a wide range of applications, such as human-computer interaction, intelligent animation synthesis, video surveillance, etc. Single camera video, due to the lack of depth information, causes difficult challenges of estimating 3D human pose. This paper proposes a modified particle swarm optimization method combined with human motion prior knowledge in order to achieve a robust analysis-via-synthesis strategy. Due to the numerous applications of human upper body movements, we are focusing on creating a front-view human upper body model.

Due to the high dimensional body configuration of human pose estimation, particle swarm optimization, with great global search ability, has a very slow convergence speed. Therefore, our modified algorithm uses annealing method so that the particles can converge faster to the lowest likelihood function value. This fact makes our algorithm more effective.

Integrated use of several image features, such as silhouette, arm silhouette, ratio silhouette area, edge, motion and skin color, constructs our cost function. Each feature has its unique purpose in order to achieve much more accurate and robust pose estimation results.

Constraining human body configuration, including the perspective scope of joint movements angle range constraints and non-penetrating constraints of limbs, is to make sure estimating human pose in the feasible region, preventing illegal pose data, and improve the accuracy of 3D human tracking. In addition, a trajectory feature is used to re-distribute particles for every frame tracking.

Experiment results show that our modified algorithm combined with cost function provides a much more accurate and robust result than downhill simplex algorithm [1] and Annealing Particle Swarm Optimization Particle Filter [2].

Key Words: Single Camera; 3D Human Upper Body Pose Estimation; Annealing Particle Swarm Optimization; Human Body Model; Human Motion Constraints

Table of Contents

List of Figures

List of Tables

Acknowledgements

During this studying time for a master's degree in the Electrical Engineering Department at University of Washington Seattle, it is a very important and meaningful part in my life.

Thanks Professor Jenq-Neng Hwang’s guidance, care and help. I learned a lot from Professor Hwang, including knowledge-wise and attitude-wise.

Thank you my lab mates, Shian-Ru Ke, Xiang Chen, Kuan-Hui Lee, Meng-che Chuang, Yong-Jin Lee, Chun-Te Chu, and Pei-An Lee. Thanks for their help, including recording videos and academic discussions.

Chapter 1:
INTRODUCTION

1.1 Research Background

Human pose estimation is a very important research direction in Computer Vision field. The purpose of pose estimation is to be able to make a deep analysis over videos, which is to obtain human motion parameters and use these parameters for semantic analysis and behavior understanding. 3D human pose estimation represents a relative position of real human body parts in 3D world space. 3D human motion recovery is to gather 3D human motion data on every frame from typical videos [3]. Some papers also call human motion recovery to be 3D human motion estimation, motion reconstruction [4], motion capture [5][6], motion tracking [7][8][9], or pose estimation [10][11]. These 3D human motion data recovered from videos are farther used for research on emotion recognition, action recognition, identity recognition, and behavior recognition, etc.

Video based 3D human motion recovery research can be used on many applications in many different research fields, such as:

Human Computer Interaction: people inputs relevant information and requests to computer machines through certain devices [12][13][14], and then computer machines will respond to those requests automatically. In people's daily life, sign language and posture body language is the most common communication ways between people except speaking language. The purpose of intelligent human computer interaction is to get rid of keyboard, mouse, and other conventional computer equipment's restriction. The goal is that computers can communicate with people spontaneously and conveniently like an ordinary person, through some natural communication methods, such as voice messages, sign language, and body language. Besides voice messages and natural human languages, computer can interacts with people more efficient with the supplement of visual information. With the applying of visual analysis, a high technic on intelligent human computer interaction, pose estimation and recognition from sign language and body language can be more reliable than voice recognition under noise environments, such as airports and factory, etc.

Video Surveillance: The ideal surveillance system [15][16][17] should be able to track human motion in real-time automatically and accurately analyze and judge human behavior through analyzed video data obtained from typical RGB camera. The real-time means that when an abnormal or a potential criminal behavior happens, the surveillance system can accurately alert people on time so that it fulfills the purpose of advance warning and effectively preventing similar events from happening. Although surveillance camera is already widely used for commercial purpose, it does not play its real-time and active monitoring role very well. Typically, company still need to hire a certain amount of securities, and this fact will cost many human resources, material resources, and financial resources. Human motion analysis has a wide range of applications, such as monitoring and warning on some related sensitive occasions like transportation hinge or main streets; analysis on the statistics of the flow of people traffic in public places; automatically monitoring on the restricted area of an exhibition or a big game occasion; and surveillance of carried items, etc. Figure 1.1-1 shows an example.

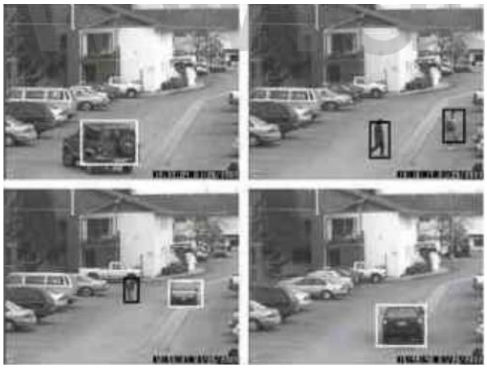


Figure 1.1-1: Example of Video Surveillance

Digital Video Game: Commercial motion capture equipment is widely used nowadays, which makes obtain human body modeling, human pose and motion parameters much easier; therefore, we can use the equipment to tell the computer to achieve very realistic simulation [18][19] on animation, character's body parts in a game, and motion and behavior interaction. For example, in move King Kong, King Kong's pose data is collected from processed real human pose data. Pose estimation from single camera has very good progress nowadays so that it is possible to obtain abundant human motion data from classic movies, sports videos, and history record videos; and then using these data for computer animation and video game development. Figure 1.1-2 shows an example.

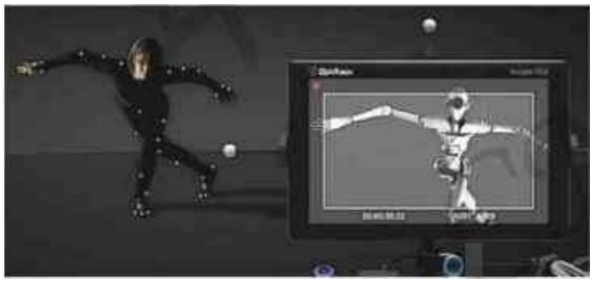


Figure 1.1-2: Example of Digital Video Game

Video Retrieval and Indexing: The rapid development of multimedia technology and the rapid expansion of the visual information is an urgent need to find a better way of effectively managing and retrieving visual information resources. Thus, more and more people pay attention to multimedia information retrieval technology based on content-based image and video; and this fact emphasizes multimedia technology and image processing become a very important research direction [20]. However, the problem is that there is no good solution to decide which feature should be used to do retrieval. In most videos, people are usually the core content, so using 3D pose estimation of the human body in the process of retrieval as search basis information can greatly improve the retrieval speed and accuracy. Furthermore, we can achieve a compression of the video data content through the modeling of the body parameters so that we can save a lot of storage space and transmission costs.

Medical Diagnosis: Current medical gait analysis is to obtain a video sequence of the movements of the patients, and then use these movement data for computer analysis and a series of comparisons with normal gait data [21]. Video-based 3D human motion capture can provide

strong support for medical diagnosis and treatment; it can assist clinical medical diagnosis of orthopedic patients, thus contributing to injuries or deformity of the judgment of some parts of the body to help doctors develop treatment programs, so that doctors can use more effective adjuvant therapy. In addition, video-based 3D human motion capture can also be used as a measure of the degree of recovery after surgery.

Video Human Motion Analysis: In video sequences, by tracking the movement of human joints or the movement trajectory of multiple people, we can obtain the motion model of the human body, interpret human's behavior, and analyze the group cooperation. This video analysis can be applied to sports video analysis, assistant athletes training to improve race performance and other aspects. For example, the acquisition of the precise human motion tracking parameters can be used in the formulation of the football team tactics and specify reasonable technical movements based on different players' characteristics. In addition, the formulation of personalized training programs in dance, golf, tennis, table tennis and other athletic events need precise athletes' motion parameters as references.

Biomechanical Analysis: Biomechanics is a science, specializing in human and animal movement and motion tracking technology is its foundation. Biomechanics need to consider the movement of the organism from the two aspects, physiology and mechanics. The structured representation of the human or animal can be obtained by the motion tracking techniques, such as the skeleton in the form of bio-movement posture. Biomechanics experts can make use of the data to be analyzed, to determine what kind of action is the best to meet the mechanical properties, and what kind of action is easy to cause harm to the organism itself.

Smart Appliances: The so-called smart appliances refer to the use of advanced technology and makes mechanical household appliances into an intelligent device. Intelligent home appliances reflect the latest technology in household appliances field. Now, a lot of intelligent household appliances, with simulation of human intelligence, can accurately understanding the behavior of captured human motion so that it can automatically make intelligent decisions.

Table 1.1-1 shows an example of briefly introducing the actual application of pose estimation.

Common Application Field	Specific Applications
Human Computer Interaction	1. Social Interface 2. Sign Language Translation 3. Control Interface of Human Pose 4. Signal Transmission under High Noise Environment
Surveillance System	1. Door Access Control 2. Parking lot 3. Supermarket, Mall 4. Vending Machine, ATM machine 5. Transportation Management
Digital Video Game	1. Virtual World Interaction

	2. Character Animation
Video Retrieval and Indexing	1. Sports Video Indexing
Medical Diagnosis	1. Clinical Study of the Surgical Patients
Motion Analysis	1. Golf, Tennis Sports Training 2. Dance Choreography and Rehearsal

Table 1.1-1: Example Applications

Although video-based 3D human motion recovery application prospects is quite extensive, but due to the presence of many difficult aspects, such as non-rigid human body description, the ambiguity of 3D human body model to the 2D projection, human body model of self-occlusion, high-dimensional state space search, image noise, the camera moves, and complex background image feature extraction and matching, making the 3D human pose data recovered from video images exist a lot of uncertainty, and therefore, 3D human pose estimation field is a challenging and concerned forefront.

Currently, motion capture devices with the mature technology are used in the interior of a single background, and require multiple cameras from different angles to simultaneously record video streams, and ask tracked object body to attach the photosensitive marker for body joint point tracking. Thus, all these facts make motion capture application is greatly limited. However, in many actual occasions, the video source is obtained by single camera, human body is not attached to any special markings, and the background is complex and changeable. The loss of the body's own self-occlusion and image depth information makes the recovered results from monocular videos exist a lot of ambiguity, so it is necessary to study the unmarked monocular video in 3D human motion recovery theories and methods.

1.2 Worldwide Research Overview

In view of the importance of human motion analysis, people start to research on video-based human motion tracking method [22] in a very early time. Early research focuses on locating the entire human body and tracking human motion contour; after 20 years of development, research now has higher requirements for video-based human motion tracking technology, and shifts the focus to 3D human motion recovery which integrated with image processing, computer vision, computer graphics, artificial intelligence, human body kinesiology, and machine learning theory, become a hot interdisciplinary area. Many important international meetings, such as ICCV, ECCV, CVPR and authoritative journals such as IJCV, CVIU, PAMI, IVC, currently have related case studies.

1.2.1 Important Research Overview In America

The United States and many countries in Europe have carried out a large number of research projects involving human motion analysis in a very early time. In 1997, the U.S. Defense Advanced Research Projects Agency associates with Carnegie Mellon University, the Massachusetts Institute of Technology, and some other universities, establish a major project of

visual monitoring called VSAM [23]. Its core content is a research for automatic interpretation of video of ordinary civilian and monitoring battlefield scene. Real-time visual monitoring system W4 is able to locate and accurately segment the human body, and achieve synchronous tracking of multiple people [24]. Most of these application projects are aimed at locating entire human body and track human as a whole, and they are not able to track the movement of the body limbs.

U.S. Brown University vision research group led by Michael J. Black has been committed to research on the model-based human motion analysis [25][26][27][5]. They use particle filter method to track model-based human body. In 2006, the research group also provides human motion datasets HumanEva [5] for open source download so that it makes the research of human motion analysis field have a public platform for comparison of different methods. These datasets are collected in the indoor environment of simple backgrounds. Therefore, we need more natural test datasets for complex context human pose estimation algorithm.

Kinect developed by Microsoft is a big breakthrough in pose estimation field. Specifically, Kinect used captured image to compare with its internal existing human body model, with the help of PrimeSense software, camera detection, and capturing user gestures. Every matched existing internal human body model will be created into the skeleton model, then the model is converted into virtual characters that role by identifying the key parts of the human skeleton model to trigger action. Kinect uses its infrared camera to obtain accurate depth information, which is a big point that troubles normal RGB camera. The Kinect group from Microsoft uses the random forest tree method to provide a very robust, accurate, and efficient real-time result of pose estimation with the help of a large training datasets and the depth information [41]. However, although the infrared camera provides key depth information for pose estimation, it is impractical to use in real world for now. Therefore, even though Kinect can provide perfect results, single camera 3D pose estimation is still a hot topic in the field. Figure 1.2.1-1 shows an example.

Figure 1.2.1-1: Example of Kinect

1.2.2 Important Research Overview outside America

Bill Trigg research group, from French National Institute of Automation, in recent years has been committed to research on video-based human motion analysis, and accomplishes some excellent results for human motion recovery in monocular video [29][30][31]. A major project of this team LEAR focuses on the robust description of the basic appearance of the shape of the human body, uses machine learning method to return motion data, and thereby restores 3D human motion [32]. One of their group members, C.Sminchisescu, goes to the University of Toronto,

Canada after graduation to continue to engage in human motion analysis studies, and now has become a representative figure in human motion analysis field [33]. This team's research focuses on human motion recovery based on silhouette features, but the silhouette is often difficult to extract because of the changes of natural background scenes or movements of the camera. Thus, to study the robustness of human contour extraction algorithm and the robustness of movement tracking method which does not depend on the outline feature is particularly important.

Particle Swarm Optimization method is also a popular method in human motion recovery method. Dr Spela Ivekovic from United Kingdoms has done many pose estimation research based on Particle Swarm Optimization method [34][35][36][37]. Her main contribution is to use modified Particle Swarm Optimization method, such as HPSO [34], to achieve robust pose estimation with entire human body or upper human body only from multi-view videos. However, her experiment only focuses on multi-view videos, and is not very useful to commercial use of single view video pose estimation analysis.

Computer Vision group in the Academy of the Swiss Confederation led by Professor Pascal FUA has been doing in-depth research on human motion analysis for many years. His student R.Urtasun uses Gaussian process latent variable model and Gaussian process dynamic model for human motion modeling for the first time [38][39][40]. His paper [40] uses Gaussian process latent variable model for modeling a variety of sports motion so that it can achieve a fusion model of different sports walking, running, jumping, etc., but the model is mainly used for the synthesis of human motion style animation. How to learn from their thoughts of motion prediction model and applied to motion tracking is worth further studying.

Institute of Automation of Chinese Academy of Sciences has made some breakthroughs in the gait recognition visual monitoring area. The development of their system can realize people’s identity automatically from a long-range distance. Zhejiang University and Microsoft Visual Perception Laboratory cooperate to develop a binocular camera video animation system, which solve the problem of high cost of obtaining animation and of obtaining human motion data that is restricted by devices. However the system needs the human body joints to be attached with marked points. It is similar to the current commercial motion capture equipment. Therefore, the system fails to capture the movement of human body in regular video streams.

Dr. Xiangyang Wang from Shanghai University proposes a new algorithm called annealed particle filter based on particle swarm optimization for pose estimation [2]. Using HumanEva datasets [5], Dr. Xiangyang Wang’s algorithm outperforms several popular algorithms with respect of accuracy and efficiency, such as traditional particle swarm optimization, or particle filter. However, his features silhouette and edge have limitations for pose estimation, which causes his results not very robust.

1.2.3 Technical Existing Circumstances

On the whole, the monocular video-based 3D human motion recovery is currently in the early stages of development. They are not able to compare with commercial multi-camera mark-based capture system whether from the view of real-time, accuracy or tracking robustness. It is easier to recover the depth information of the body movement using a plurality of cameras, so many 3D

pose recovery researches work based on multi-camera. However, the real existence of the video source is often single camera. Thus, in recent years, monocular video-based 3D human motion recovery gradually attracted researchers' attention. This research is still in laboratorial stage, but with the development of computer technology and the improvement of the related creative theory, unmarked ordinary monocular video-based 3D human motion recovery technique will gradually goes mature.

Currently, 3D human motion recovery in monocular video mainly uses top-down generation approaches and bottom-up discriminant approaches, and particle swarm optimization based method is a widely used framework of current generation approaches. Particle swarm optimization algorithm initializes the particle swarm, including its random position and velocity, and then calculates the fitness value or cost value of each particle, and then through an iterative search for the optimal solution for each particle by tracking two "extreme" to update their direction of movement. When there is a good enough fitness value or when the maximum number of iterations is reached, the algorithm terminates.

1.3 Research Work about This Paper

This paper's 3D motion data is obtained from only large-scale human movement, upper body to be specific, not including the small-scale motion, such as the fingers, facial expressions, etc.; video source is from ordinary camera, in which the human body is not required to wear special clothing or wear special mark; background has no special requirements.

This paper studies monocular video-based 3D human upper body pose estimation using annealing particle swarm optimization algorithm, including:

1. Add the annealing idea in the traditional particle swarm optimization algorithm so that it can adapt the high dimensional space search of 3D human motion and improve tracking efficiency.

2. Add join angle constraints and non-penetrating constraints into the whole two human motion tracking process in order to achieve more accurate tracking performance than existing tracking algorithm.

3. Select silhouette, edge, motion, skin, restricted arm silhouette, and ratio silhouette as the six features, and use these feature images compared with projection images obtained from our predicted model to form match likelihood. Finally, this cost value is used as fitness value for every particle in the particle swarm.

4. Add simple motion prediction data in the process of pose estimation.

5. Verified experiment results by compare with other people's method [1][2].

6. The design and implementation of 3D upper human body pose estimation based on annealing particle swarm optimization system.

1.4 Paper Organization

In this thesis, the main research is about monocular video annealing particle swarm optimization based 3D human motion tracking. The entire paper discusses as the following:

The first chapter describes the significance of the study of 3D human pose estimation, introduces the development history and current circumstance in this field, and briefly discusses the importance and challenge of the research of 3D human pose estimation using single camera.

The second chapter briefly describes several features of the human body image, several types of model representation related to 3D human body, including skeleton model, shape model, and three kinds of 3D to 2D projection model. This chapter also introduces the Euler angles of human skeleton movement.

The third chapter details the whole process of annealing particle swarm optimization based 3D motion tracking, including the idea of annealing particle swarm optimization algorithm, the setup of human motion constraints, simple use of prediction trajectory, the establishment of the cost function, and finally the design and implementation details of the system of the annealing particle swarm optimization based 3D motion tracking.

The fourth chapter introduces two other popular methods first, and then compares the results from different adjustable parameters, and at last analyzes the experimental result.

The fifth chapter summarizes the work done in this article, and gives the direction of further research work.

Chapter 2:
3D HUMAN BODY MODEL AND IMAGE FEATURES

2.1 Outline

The purposes of the body movement tracking are to obtain continuous body position and pose estimation data from image sequences. The system accuracy is influenced by adverse factors, such as the possible self-occlusion of different body limbs during the tracking process, the different look of the appearance from different perspectives, and relative disordered background in the actual tracking environment, and therefore, modeling the 3D human body seems very necessary.

The human body is a very complex system composed of more than 200 joints. It requires all the joints data to estimate human pose movement. This fact makes the realistic simulation of human motion is more complex than some average rigid body. In order to reduce the difficulty of motion tracking, researchers need to consider the use of the human body model to introduce the necessary prior information of human motion and abstract and simplify the representation of the real human body in a certain degree so that they can simulate the real human motion.

Usually a human body model includes three basic components: point, line and shape. Point used to indicate the position of the joints of the human body; the line is used to connect points between each joint of the skeleton structure; the shape is the geometric model that attached to the skeleton structure, which used to represent muscle or skin tissue. Figure 2.1-1 is an example of 3D human body model.

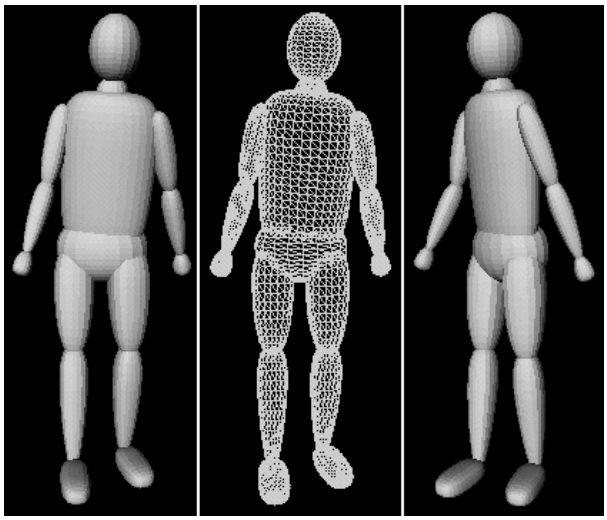


Figure 2.1-1 Human Body Model

Commonly used 3D human body models are skeleton model and shape model. The skeleton model defines the kinematic properties of the human body model, such as the various parts of the angle, length, etc.; shape model defines the appearance characteristics of the human body. When the real human body is indicated by the use of parameterized model, the pose of the human body is a point in motion parameter space. Through a nonlinear projection model from 3D space to 2D, 2D image can be created by projecting 3D model parameters. Then, use the 2D image features to

match the human body model after the nonlinear projection so that finding the most appropriate 3D body pose parameters as the final 3D human body tracking results.

When reasonable human body model is constructed, researchers can use kinematics and biomechanics of the human body constraints to constrain the pose parameters; and because of the guidance of the geometric model based on real human body, the whole tracking process is also more in line with the real situation. This is mainly manifested in two aspects: First, be able to judge the self-occlusion situation, and then make special treatments. Second, exclude some error tracking parameters to improve the reliability of the tracking results by combining with the characteristics of kinesiology and biomechanics knowledge. In addition, after obtaining each joint data of the body movement, researchers can further improve the system from the skeleton, muscles, and skin level so that they can achieve high realistic human body simulation.

2.2 Human Skeleton Model and Representation

Skeleton model is a representation of the topological structure and shape of the human body. It is formed by some thin lines, and it can effectively reflect the connectivity and hierarchy of human shape. Therefore, the human skeleton model contains a wealth of sports information. It can be used both for the identification of human characteristics, and also be used for the real-time tracking of the human body in motion.

2.2.1 Joint Model

The joint is position that used to connect to the various parts of the human body. The human skeleton model is composed of a number of rigid bodies linked together through the joint. The hierarchy of these joints is used to abstractly represent the structure of the human body. Because joint reflects the relationship between the various parts of the body, the joint model is the basis for building a complete human body model. The pose of the human skeleton model is controlled by all the joint angles. Defined joint model can use the relative movement of the joint connection between the rigid bodies to describe as some of the motion parameters and joint constraints of the parameter values. The purpose of introducing the constraint is to control the motion range of joint within an approximate real space. The degree of freedom of the joint is defined as the number of independent variables that are needed to provide a complete structure of human body pose. The more degrees of freedom of the joint, the more flexible the movement will be.

The length of the various parts of the human skeleton and scope of activities of the joint varies for different people, but the real characteristics of the joint can be simplified by defining an ideal joint model. There are three commonly used joint models.

1. Rotation joint model: This joint is only one degree of freedom (DOF), and the angle of rotation is limited to within a certain range.

2. Elbow model: The elbow model contains two independent degrees of freedom. It is able to simulate the limbs around the twisting motion and flexion and extension movements, such as elbow or knee.

3. Ball shape joint model: This joint model contains three rotational degrees of freedom; one degree of freedom is for the rotational angle around the rotation axis; the other two degrees of freedom to determine the direction of the axis. Thus, ball shape joint model can simulate structure like the shoulder joint.

2.2.2 Skeleton Model

Through a tree structure, a human skeleton model can be represented of combining all the joints of the body. Each node in the tree structure represents a joint. In the structure, there is strict parent-child relationship between the nodes. Except the root node, every node has a parent node, and all nodes have a common ancestor is known as the root node. The position of each node is based on the parent node, and their relationship is represented by a local coordinate system transformation. Figure 2.2.2-1 shows an example of skeleton model.

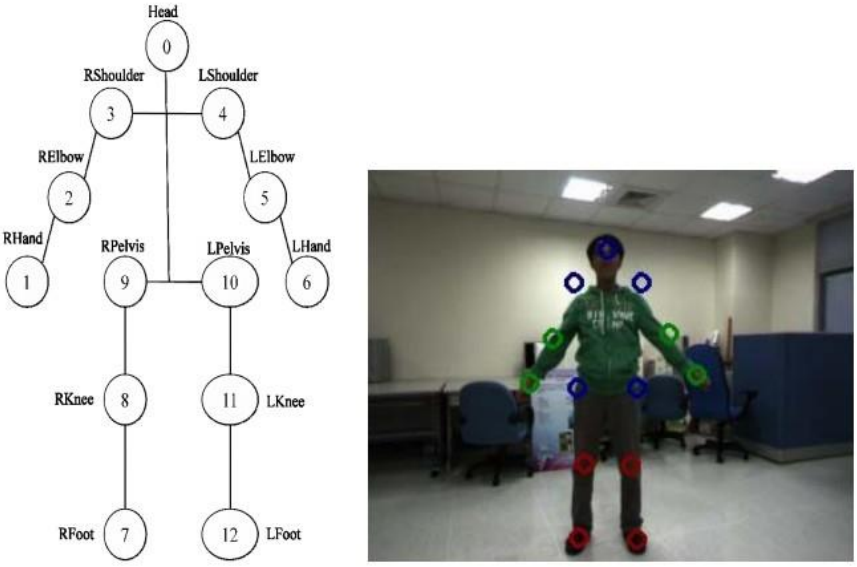


Figure 2.2.2-1: Skeleton Model

When giving a specific definition of the human skeleton model, the appropriate human body model [42] should be chosen based on the actual application background. According to actual needs to define the type of the human skeleton model, the goal is to achieve a balance between the accuracy of model representation and the computational complexity. In the parameterization process, in order to reduce the opportunity to have a false pose, the number of parameters should be controlled to be no more than the actual number of degrees of freedom. A good model should meet the actual needs of the case of leading into the least number of degree of freedom.

2.2.3 Euler Angles Description

In an angle representation system for 3D motion data, Euler angles and coordinate system transformation is widely used. Using Euler angles, researchers can obtain a coordinate system from another coordinate system through three rotations, such as conversion between father and son coordinate system. Euler angles are a kinematics concept. It is used to uniquely identify the sentinel rotating rigid body position of a group of three independent angular parameters,

indicated by yaw angle, pitch angle, and roll angle, as Figure 2.2.3-1 shows. Another Figure 2.2.3-2 gives an explanation of these three angles. In Figure 2.2.3-2, angle α is the angle between the *x*-axis and the *N*-axis, which implies a rotation around the *z*-axis; β is the angle between the *z*-axis and the *Z*-axis, which implies a rotation around the *N*-axis; γ is the angle between the *N*-axis and the *X*-axis, which implies a rotation around the *Z*-axis.

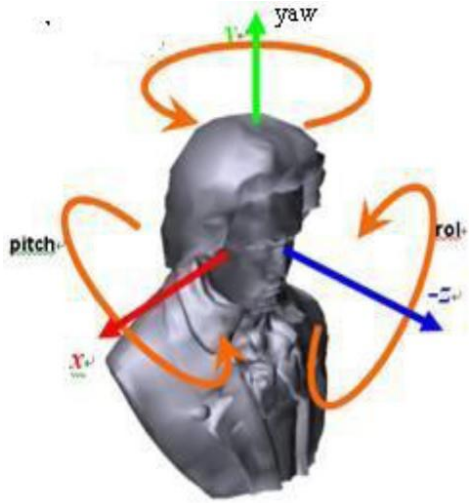


Figure 2.2.3-1: Euler Angle Representation

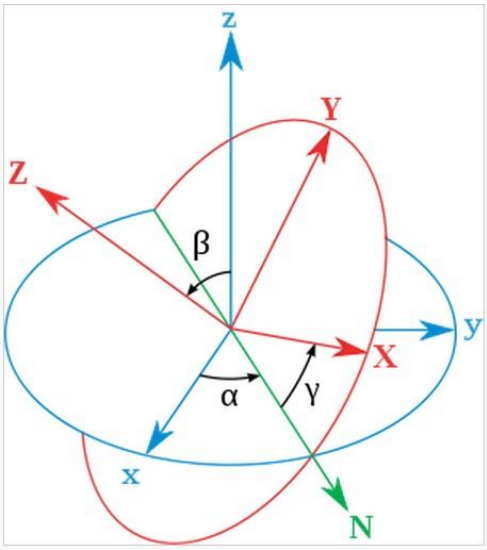


Figure 2.2.3-2: Euler Angle Representation -- the xyz (original) system is shown in blue, the XYZ (rotated) system is shown in red. The line of nodes labeled N, is shown in green

Euler angles are a system of rotating around a certain axis. However, there is a certain rule about the order of the spinning, such as rotating around an axis, and then rotating around another axis for other angle values. Euler angle rotation sequence can be Z--X--Y, X--Y--Z, X--Z--Y, etc. In BVH file, this sequence is defined as Z--X--Y. In ASF/AMC files, this sequence is defined as X--Y--Z.

Each Euler angle is a conversion based on its rotation axis from one coordinate system to another coordinate system. The basic method of coordinate transformation is that use one point multiply by a rotation matrix between the two coordinate systems. In the coordinate changes, the following matrix will be used:

Transformation Matrix: See formula Figure 2.2.3-3. The t_x, t_y, t_z is the shift amount of the new coordinate system with respect to the original coordinate system.

$$T(t_x, t_y, t_z) = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Figure 2.2.3-3: Transformation Matrix

The three rotation matrices around the x, y and z axis are formula Figure 2.2.3-4, and γ, θ, and β are the rotation angles of x-axis, y-axis, and z-axis.

$$R_x(r) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos r & -\sin r & 0 \\ 0 & \sin r & \cos r & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_y(\theta) = \begin{pmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$R_z(\beta) = \begin{pmatrix} \cos\beta & -\sin\beta & 0 & 0 \\ \sin\beta & \cos\beta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Figure 2.2.3-4: Rotation Matrices

For example, for a point p(x,y,z) in a sub coordinate, its parent coordinate position is P(X,Y,Z). Then, the point p(x,y,z) to point is P(X,Y,Z) is calculated as formula Figure 2.2.3-5:

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = T(t_x, t_y, t_z) * R_z(\beta) * R_y(\theta) * R_x(r) * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Figure 2.2.3-5: Example

Because the tree structure to represent the relationship between the skeletons and the parent-child relationship between adjacent nodes of the tree structure model, the pose parameter values of each body parts are usually based on the child – parent coordinate system.

2.3 Human Body Shape Model and its Representation

The shape model is used to describe the appearance of the human body, which is a model that attached to the skeleton model. In the process of motion tracking, researchers mainly use the projection image from 3D pose estimation to calculate the cost function. Shape model can be divided into three types, as the following presents.

The first type is an appearance model of using abstract geometry to describe the human body parts. Figure 2.3-1 shows an example of circular cylinder model [31][43]. Figure 2.3-2 shows another example of truncated cone model [28][44]. In addition, there are ellipsoid model to represent human body limbs.

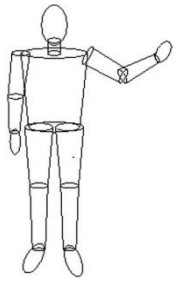


Figure 2.3-1: Cylinder Model

Figure 2.3-2: Truncated Cone Model

The second is the use of simulation technology combined with computer graphics principles of the human body shape model, such as Figure 2.3-3 shows [45]. Establishing the acicular model on the human skeleton; one end of the needle is attached to the skeleton; on the other end is used

for the reconstruction of the skin. The model reflects the continuous deformation of the human skin. However, the model needs to locate joint through the sign points and by using expensive data-obtaining equipment.

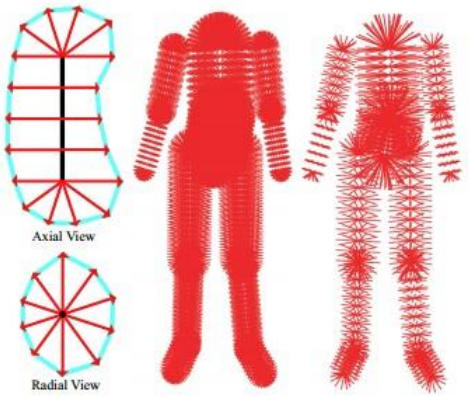


Figure 2.3-3: Acicular Model

The third kind is the body surface model, such as model the human body using convolution curve surface [46], also as Figure 2.3-4 shows. In order to optimize the tracking process, the analytic representation of the human body projection is provided after these model explanations.

Figure 2.3-4: Convolution Curve Model

2.4 Projection Model

By the principle of camera imaging, each point in 3D space can be transformed into a point of 2D image using an appropriate projection model. In order to get these correspondences, camera parameters is necessary to be obtained. In computer vision, researchers usually use camera calibration method to get these parameters. Commonly used projection model includes the following three:

1. Perspective projection model: A perspective projection creates 2D images of 3D objects by projecting lines from a center of projection through an image plane until they meet the objects. It has several characteristics, such as the use of vanishing point, the use of the distance between the center of projection and the image plane, and regular changes through same size objects. These

facts can show a space image that truly manifesting the object image. Perspective projection principle conforms to people's psychological habit, which is the size of the object is related to the distance from focus. For an arbitrary point P in the global coordinate system P(X, Y, Z), there is a point in the image plane p(x, y, z). The camera focal length f satisfies the formula Figure 2.4-1. Figure 2.4-2 shows an example of perspective projection model.

$$x = f\frac{X}{Z}, y = f\frac{Y}{Z}, z = f$$

Figure 2.4-1: Formula of Projection

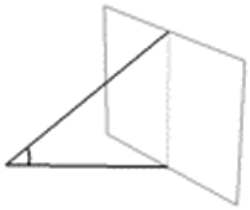


Figure 2.4-2: Example of Perspective Projection

2. Orthogonal projection model: Orthogonal projection is parallel projection. All projection direction is parallel and no intersection point. It is a form of parallel projection, where all the projection lines are orthogonal to the projection plane, resulting in every plane of the scene appearing in affine transformation on the viewing surface. For an arbitrary point P in the global coordinate system P(X, Y, Z), there is a point in the image plane p(x, y, z). The camera focal length f satisfies the formula Figure 2.4-3. Figure 2.4-4 shows an example of orthogonal projection model.

$$x = X, y = Y, z = Z$$

Figure 2.4-3: Formula of Projection

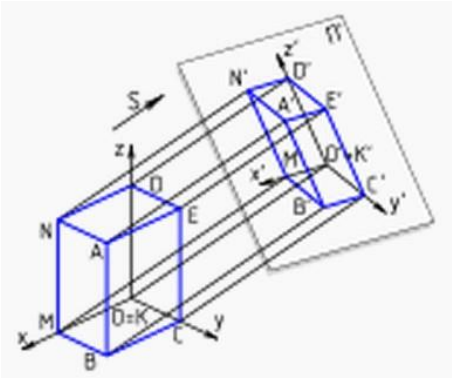


Figure 2.4-4: Example of Orthogonal Projection

3. Weak Perspective Projection Model (Variable Scale Orthogonal Projection Model): Weak perspective projection is an approximation of the perspective projection. In fact, it is a scaled

orthographic projection: first, the object is projected onto the image plane by a set of parallel rays orthogonal to the plane; second, the image of the object is scaled. This approximation works if the object is close to the optical axis of the camera or its dimensions are small relative to the distance from the camera. For an arbitrary point P in the global coordinate system P(X, Y, Z), there is a point in the image plane p(x, y, z). The camera focal length f satisfies the formula Figure 2.4-5, where Z is the physical center of mass position in the global coordinate system. Figure 2.4-6 shows an example of weak perspective projection model.

$$x = \frac{X}{Z}, y = f\frac{Y}{Z}, z = f$$

Figure 2.4-5: Formula of Projection

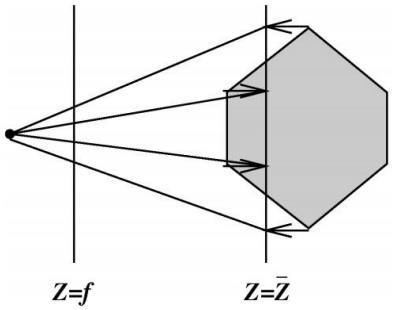


Figure 2.4-6: Example of Weak Perspective Projection

2.5 Image Features

In the analysis study of 3D human motion based on video images, a very important part is that the accurate image features that are extracted from the video image. According to the basic theory of pattern recognition, feature is a selected measured way; it should express the characteristics of objects as much as possible, but also try to avoid introducing too much redundant information. In the process of human motion tracking, it is very important to select the appropriate image features and establish the cost function by comparing projected pose features and true pose features. In order to correctly measure the degree of matching, researchers need comprehensive consideration of various factors to select the appropriate image features so that researchers can take advantage of two pose feature data's degree of similarity to measure the degree of matching. The image is just a collection of a group of pixels. It does not directly reflect the human pose but contains a lot of information. For video sequence, researchers should go through image processing to extract an accurate description of the image feature information of the human pose. In the field of image processing, pattern recognition and computer vision, the most common features of constructing the cost function are contours, edges, silhouette, motion, skin color, and optical flow.

1. Silhouette: The silhouette is the human region in an image, and uses to match with the projection area of the model. Compared with the edge feature, silhouette feature is relatively less susceptible to external noise, but it is easy to lose specific details of the human body parts. Silhouette feature can be obtained by reducing the background and morphological filtering approach. Figure 2.5-1 shows an example of silhouette.

Figure 2.5-1: Example of Silhouette

2. Edge: The edge is the image region that having a larger gray-scale difference. It is one of the most basic visual characteristics of people can see. However, the precise detection of the edge of the complex background is not easy. Edge feature of the human body can be extracted by making use of similar gray-scale value of inner pixels of the same objects and the obvious difference of the gray-scale value of different objects. Figure 2.5-2 shows an example of edge.

Figure 2.5-2: Example of Edge

3. Skin color: The skin color feature is a description for human skin by defining pixels whose (R-G) value is between 20 and 80 in a RGB video sequence. The use of skin color is to help extremely accurately track human pose, especially for head or hand, since they are the human skin that typically expose in videos. The disadvantage is that skin color is not invariant to illumination and also some RGB color, such as red, that has similar range of skin color; and therefore this fact will affect the tracking process. Figure 2.5-3 shows an example of skin color.

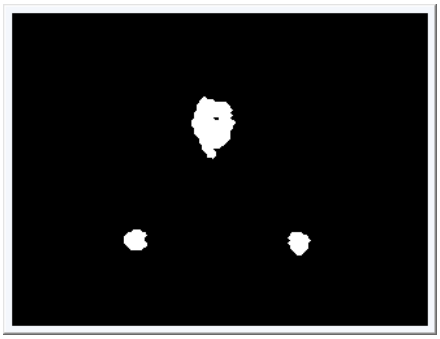


Figure 2.5-3: Example of Skin Color

4. Motion: The motion feature is the frame difference for two consecutive frames in a video sequence. It serves three main advantages. First, in a human body analysis video, human beings typically are the only moving objects, and therefore, frame difference represents the moving human body parts and nothing else. This fact means that motion feature can reduce the effect of noise background, which can help enhance the pose estimation accuracy by combining with silhouette feature or other features researchers choose to use. Second, motion feature can help self-occlusion. For example, if a person's arm gets across with his/her own torso; in a silhouette image feature, the torso part blocks the arm's silhouette, which makes tracking human arm based on silhouette impossible. In the meantime, the torso does not move, so the motion feature can provide a clear binary image for human arm to help self-occlusion problem, like Figure 2.5-4 shows. Third, motion feature can increase the efficiency of human pose estimation. For instance, when human torso does not move, the motion feature can help to decide the algorithm not to search for torso pose estimation. Therefore, this fact helps increase the efficiency. The disadvantage of motion feature is that as Figure 2.5-4 shows, the binary image cannot represent the whole human body, like silhouette does. Thus, researchers choose to use motion feature combined with other features to estimate human pose.

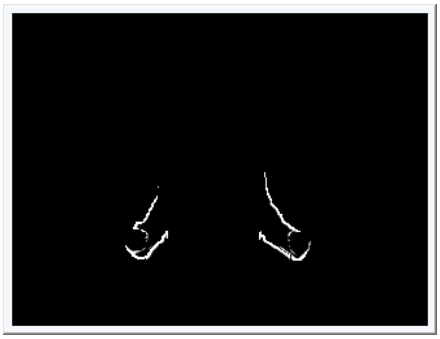


Figure 2.5-4: Example of Motion

5. Contour: The contour is the border of the silhouette, and it is another type of an edge feature. Contour is more robust than edge feature since it comes from silhouette, which means it goes through background elimination already, but it is relatively computational expensive. Figure 2.5-5 shows an example of contour.

. Figure 2.5-5: Example of Contour

6. Optical Flow: Optical flow [47] is a time-series data, which reflects inter-frame points or consistent movement of features. Optical flow extraction does not require background fixed, but it is computationally intensive and poor noise immunity. The method can be used to estimate model movement parameters [48]. Figure 2.5-6 shows an example of optical flow.

Figure 2.5-6: Example of Optical Flow

2.6 Chapter Summary

This chapter describes some of the commonly used 3D human body model and its representation. It firstly introduces the definition of the human skeleton model and common representation, and then introduces a joint model and skeleton hierarchical model. In addition, this chapter introduces human body shape model and its representation, and simply introduces silhouette, edge, motion, skin color, contours, and optical flow; some commonly and some not commonly used special human body image features. Then, it presents the advantages and disadvantages of using those features to construct the cost function. This chapter also elucidates the Euler angle principle that will be used for estimating 3D human pose, and gives specific form of the rotation matrix and the coordinate conversion formula. Conversion of the Euler angles is a very important theory for 3D space operations, and is frequently applied to 3D data operations. It describes the orthogonal projection model, perspective projection model, and weak perspective projection model in detail for the three models of human 3D to 2D projection, and provides specific projection formulas. This chapter, as a preparation section for 3D human motion recovery basic knowledge, introduces some basic concepts and mathematical expressions of 3D human body; and provides the basic conceptual preparation for 3D human pose estimation in later chapters.

Chapter 3:
SYSTEM DESIGN AND IMPLEMENTATION of 3D HUMAN POSE ESTIMATION BASED ON ANNEALING PARTICLE SWARM OPTIMIZATION

3.1 Description of Particle Swarm Optimization

Particle swarm optimization (PSO) algorithm proposed by Kennedy and Eberhart in 1995 [49] based on swarm intelligence evolutional computing technology. The algorithm is a simulation of birds, schools of fish, and human social behavior. Its basic searching idea is that the neighborhood of the optimum particles is a high probability location of high cost value (finding max) or low cost value (finding min). Therefore, distributing more particles around the neighborhood of the optimum particles is used to enhance the searching ability of this algorithm. PSO, similar to other genetic algorithms, is a population-based optimization tools. Every particle tries not to blindly fall into local best particles through other useful information. Meanwhile, PSO also uses a random search strategy, which is different from the genetic algorithm, so PSO shows the advantages of better search performance and simple implementation when it comes to solve certain problems.

3.1.1 Motivation and Basic Idea of Particle Swarm Optimization Algorithm

The motivation of PSO is to imitate the behavior of groups of birds. Initially, Reynolds [50] puts more energy to focus on the aesthetics of the flight of birds and the laws of those birds can suddenly changing direction, dispersion, or gathering. These methods are excessively dependent on the theory of maintaining the optimal distance of the individual in the group to achieve synchronization of the flock through bird's efforts.

However, social biologist E. O. Wilson draws the following conclusions through the study of fish stocks social behavior [51]. At least in theory, in the process of searching food, no matter how unpredictable the distribution of food, individuals of group can benefit from the discovery and previous searching food experience of all the other individuals in the group; such benefits can be decisive, and exceed the harm that is brought from the competition for food. The concept of social information sharing is the basis for the development of the PSO algorithm.

Another motivation is to mimic human social behavior. An important difference between a group of humans and a group of birds or fish is that in order to avoid predators, birds or fish tend to change their physical movement for looking for food, spouse, or finding a suitable living environment; however, humans will not only adjust their physical movements, but also improve their awareness and experience. People always tend to adjust their beliefs and attitudes in order to be consistent with social elite.

PSO belongs to evolutionary computation areas. Evolutionary algorithm uses a simple binary coding technique to represent a variety of complex structures; and uses genetic manipulation and natural selection of survival of the best to do supervised learning and determine the direction of the search. Because this population searching method of the evolutionary computation can search multiple subspaces at the same time, and its characteristics of self-organizing, self-adaptive, etc., this evolutionary computation has high efficiency and versatility. PSO algorithm continues the

concept of population and uses the particle (individual) encoding method that is similar to evolutionary computation algorithm. However, the operation of PSO operation is simple, easy to implement, and high efficiency. These facts show strong vitality of the PSO algorithm.

3.1.2 Basic Particle Swarm Optimization Algorithm

In the PSO algorithm, particle swarm is formed by m particles. The position of each particle position represents potential solutions of optimization problems in D-dimensional search space. The algorithm uses the objective function to calculate every particle's fitness value based on its current position. In addition, another variable particle speed determines the flight direction and distance of every particle in D-dimensional search space. Particles update their status in accordance with the following three principles:

1. Maintain their own inertia.

2. Change position based on their own historical best position (local best position).

3. Change position based on globally optimal position of the population (global best position).

As previously described, PSO algorithm is inspired from the behavior of biological populations and uses to find the optimal solution of a problem. Each potential solution of the optimization problem can be thought of as a point in D-dimensional search space, and in PSO algorithm, each particle corresponds to such a point. These particles flight in a D-dimensional search space with a certain speed. This speed is dynamically adjusted according to its own flying experience and other particles' flying experience in the group. All particles have a fitness value that is decided by the cost function. Fitness value is used to measure the pros and cons of the particle position in the search space. Particle has memory, and can remember its own experienced best position, and also can obtain the information of global optimum position from its population.

Optimization searching method is carried out by a swarm of a group of random initialized particles in an iterative manner.

In a D-dimensional search space, there is a swarm of m particles. The i-th particle among the swarm represents a D-dimensional vector, $X_i = (X_{i1}, X_{i2}, \ldots, X_{iD})$, where i = 1, 2, …, m. Calculate i-th particle's (X_i) fitness value $f(X_i)$ through the cost function $f(X)$, and then measure the pros and cons of X_i based on the value of $f(X_i)$. The i-th particle's flying speed is also a D-dimensional vector, $V_i = (V_{i1}, V_{i2}, \ldots, V_{iD})$. Record the local best position of the i-th particle to be $P_i = (P_{i1}, P_{i2}, \ldots, P_{iD})$ and the global best position to be $P_g = (P_{g1}, P_{g2}, \ldots, P_{gD})$.

The original PSO algorithm [49] uses the following equations to update particle's status:

$$V_{id} = V_{id} + c_1 * r_1 * (P_{id} - X_{id}) + c_2 * r_2 * (P_{gd} - X_{id}) \qquad \text{Equation: 3.1.2 – 1}$$

$$X_{id} = X_{id} + V_{id} \qquad \text{Equation: 3.1.2 – 2}$$

Where c1, and c2 are non-negative constant numbers, r1 and r2 are random numbers between [0, 1], d = 1, 2, …, D, and $V_{id} \in [V_{min}, V_{max}]$, where V_{min} and V_{max} are the minimum speed value and maximum speed value. In addition, the first part from Equation 3.1.2 -1 is called "inertial component", which represents the momentum when a particle flies; the second part is called "cognitive component", which represents the ability of particle itself; the third part is called "social component", which represents the cooperation among particles. The iteration termination conditions are depending on the optimization problems. The general ways of terminating the iterations are stopping at the maximum number of iterations, or the fitness value of the global best position searched by the particle swarm is less than the predetermined minimum fitness threshold.

Paper [52] improves the original PSO method, which modified the Equation 3.1.2 -1 to be:

$$V_{id} = w * V_{id} + c_1 * r_1 * (P_{id} - X_{id}) + c_2 * r_2 * (P_{gd} - X_{id}) \qquad \text{Equation: 3.1.2 – 3}$$

Where w is also a non-negative number and is called inertia factor.

From paper [53], people research on the role played by the inertia factor. Obviously, when w is larger, the first component in the Equation 3.1.2 – 3 increases so that this part's speed becomes larger, which is conductive to jump out of a local minimum; and while w is smaller, it is conductive for the convergence of the algorithm. Therefore, researchers propose the annealing w-adjustment strategy – Annealing PSO (APSO). With the iteration of the algorithm increases, the value of w is linearly decreasing. This algorithm has been widely used, and many scholars believe this APSO is the basic PSO algorithm. In this paper, our system uses APSO algorithm.

Pseudo code for APSO:

Algorithm: APSO
Step 1: Initialize every particle i in the swarm, where i = 1, 2, ..., m

1. Initialize every particle's position *X[i]* using uniform distribution or Gaussian distribution
2. Initialize every particle's speed *V[i]* using uniform distribution or Gaussian distribution
3. Calculate every particle's fitness value P_i based on cost function, and use these values to be the initial local best fitness value for every particle
4. Compare all the fitness value, and find the gloabl best fitness value P_{gi}

Step 2: Start iteration and stop until it meets the criteria

1. Update w, c1, c2 based on Equation 3.1.2 - 4

$$w = c1 = c2 = c0 * e^{(1-\frac{m}{M})} \qquad \text{Equation 3.1.2 - 4}$$

Where c0 is the annealing factor and its value is depend on the situation, m is the current iteration number, and M is the maximum iteration number.
2. Update particle's position based Equation 3.1.2 -3 and Equation 3.1.2 - 2, and calculate the new corresponding fitness value, if $f[i] < P_i$, then $P_i = f[i]$; if $P_i < P_{gi}$, then $P_{gi} = P_i$.

3.2 3D Human Body Model in this Paper

3.2.1 Skeleton Model

Skeleton model parameters corresponds to the motion status of human body at the time, so in order to be able to achieve realistic simulation of the human body and reduce the probability of a false pose, while considering the time efficiency of the whole tracking process, the human skeleton model is defined as shown in Figure 3.2.1-1 contains 7 joints of the skeleton structure, since this paper is analyzing the upper human body pose only.

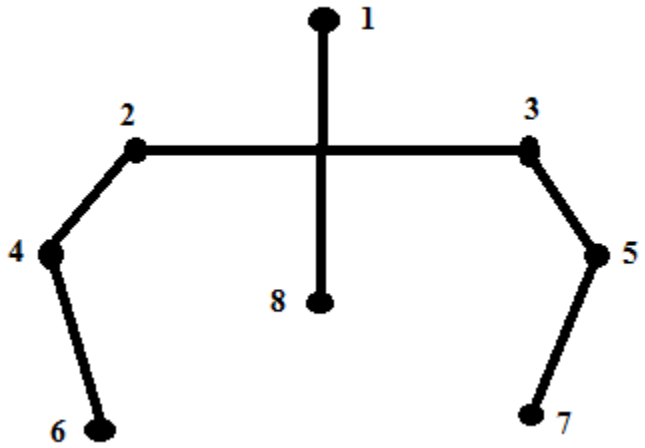


Figure 3.2.1-1: Skeleton Model in this Paper

Table 3.2.1 - 1 shows correspondence between the number of 7 joints in the skeleton model and their real joint names, also their parent nodes number.

Number of the Joint	Real Representation	Parent Node
1	head	2,3
2	Right shoulder	8
3	Left shoulder	8
4	Right elbow	2
5	Left elbow	3
6	Right hand	4
7	Left hand	5
8	Centroid of human upper body	

Table 3.2.1 - 1: Relationship of number of the joints and their real joint names & their parent nodes number

Table 3.2.1 - 2 shows the name of each skeleton model, their containing joint points, and their degree of freedom (DOF).

Skeleton	Contained Joints	DOF
Head	1	
Torso	2,3	3
Right Arm	2,4,6	5
Left Arm	3,5,7	5

Table 3.2.1 - 2: Skeleton Model's Containing Joint Points and their DOF

As in chapter 2.2.3 descried, this paper uses Euler's angle to transform the skeleton model, and therefore, variable $X_{Torso} = \{y,x,z\}$, $X_{RArm} = \{y_{up},x_{up},z_{up},x_{down},y_{down}\}$, and $X_{LArm} = \{y_{up},x_{up},z_{up},x_{down},y_{down}\}$. Table 3.2.1 - 3 explains the meaning of these variables. Variable {y,x,z} represents the three angles $\{\theta, \gamma, \beta\}$ that need to be used in the Euler's angle rotation matrices, as formula Figure 2.2.3 - 4 shows.

Variable belongs to	Variable name	Meaning
X_{Torso}	y	θ angle in formula Figure 2.2.3 – 4, where centroid is the parent node to decide the Torso matrix through formula Figure 2.2.3 – 3 and Figure 2.2.3 – 4 as example Figure 2.2.3 – 5 shows.
	x	γ angle for Torso matrix as mentioned above
	z	β angle for Torso matrix as mentioned above
X_{RArm}	y_{up}	θ angle in formula Figure 2.2.3 – 4, where right shoulder is the parent node to decide the upper arm matrix through formula Figure 2.2.3 – 3 and Figure 2.2.3 – 4 as example Figure 2.2.3 – 5 shows.
	x_{up}	γ angle for the upper arm matrix as mentioned above
	z_{up}	β angle for the upper arm matrix as mentioned above
	x_{down}	γ angle in formula Figure 2.2.3 – 4, where right elbow is the parent node to decide the lower arm matrix through formula Figure 2.2.3 – 3 and Figure 2.2.3 – 4 as example Figure

		2.2.3 – 5 shows.
	y_{down}	θ angle for the lower arm matrix as mentioned above
X_{LArm}	y_{up}	θ angle in formula Figure 2.2.3 – 4, where left shoulder is the parent node to decide the upper arm matrix through formula Figure 2.2.3 – 3 and Figure 2.2.3 – 4 as example Figure 2.2.3 – 5 shows.
	x_{up}	γ angle for the upper arm matrix as mentioned above
	z_{up}	β angle for the upper arm matrix as mentioned above
	x_{down}	γ angle in formula Figure 2.2.3 – 4, where left elbow is the parent node to decide the lower arm matrix through formula Figure 2.2.3 – 3 and Figure 2.2.3 – 4 as example Figure 2.2.3 – 5 shows.
	y_{down}	θ angle for the lower arm matrix as mentioned above

Table 3.2.1 – 3: Meaning of the Variables

3.2.2 Shape Model

After defined human skeleton model and as section 2.3 described before, in order to facilitate the tracking process, this paper uses 3D human pose data to project to the image plane and calculate the cost value through cost function based on projected features. Select the right upper arm, right lower arm, left upper arm, and left lower arm to be represented by truncated cone shape model, as Figure 3.2.2 – 1 and Figure 3.2.2 – 2 shows.

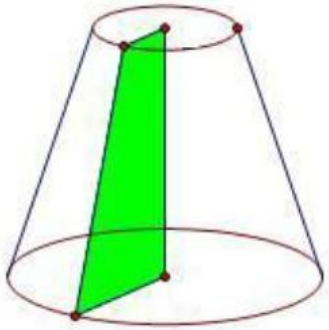


Figure 3.2.2 – 1: Truncated Cone Shape Model

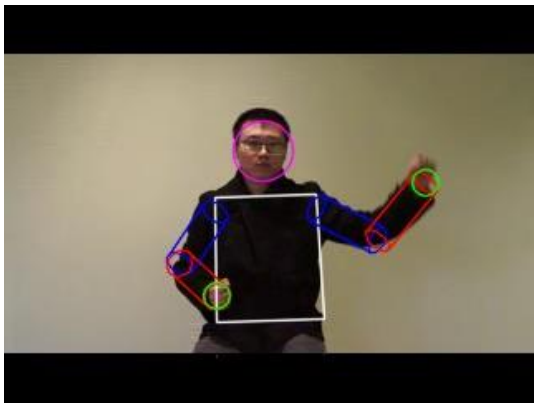


Figure 3.2.2 – 2: Example of this Paper's Shape Model

3.3 Preparation for Human Detection

3.3.1 Front View and Face Detection

The designed system is built for front-view upper human body pose estimation. Since it is front-view, the video sources will always have a face feature. Therefore, our system uses face detection as one indicator to decide if there is a human or not in the video sources. In face detection, a set of haar-like features can be used to encode the contrasts represented by a human face and their spatial relationship. The so-called haar-like features mean that they are calculated similar to the coefficients in haar wavelet transforms. A cascade of boosted classifiers working with haar-like features is trained with a few hundreds of sample faces. The boosting techniques can be chosen from one of the four, Discrete Adaboost, Real Adaboost, Gentle Adaboost and Logitboost. After classifiers are trained, they can be applied to a region of interest (searching window) in the input video source to detect human faces.

3.3.2 Initial Pose Configuration

Another indicator of deciding if there is a human is that finding three valid skin regions for head and two hands from a required initial pose. The skin regions are segmented based on skin color feature that will be introduced in the next section. The three largest areas will be considered valid and selected as candidates for head and two hands. Then, compute the distance between head and two hands, d1 for head to left hand and d2 for head to right hand. The distance ratio r_d is computed by Equation 3.3.1 - 1 so that our system can check for possible geometrical configuration of the initial head-hands position. Next, compute the distance d3 between two hands. If r_d and d3 both lie within predefined range (depend on situations), our system finds three valid head-hands blobs.

$$r_d = \frac{\min(d1,d2)}{\max(d1,d2)} \qquad \text{Equation 3.3.1 - 1}$$

3.4 Image Feature Extraction and Cost Function

Cost value is a very core part in the design of the implementation of APSO algorithm. Its purpose is to measure the degree of match between the predicted value and the real data through

the cost function. When selecting the appropriate image features to construct the cost function, not only have to consider the accuracy of the match, but also consider the simplicity of feature calculations. In addition, in order to meet the real-time requirements, the amount of feature calculation is limited within an acceptable range.

There are many features can be used to design the matching relationship between the human body model and the image, but they all have their respective limitations. How to choose the correct image feature to get a better cost function is a problem that needs to be considered. In order to get more robust tracking results, the integration of a variety of features is an inevitable choice. For example, paper [54] uses edge and illumination information for cost function; paper [55] builds the cost function based on silhouette and edge information. For the human body tracking, the problem is that not only the large amount of degree of freedom of the human body model, but also the serious bad self- occlusion; using only one feature is difficult to properly understand the image content. In theory, the more types of selected features, the smaller feature matching of the error will be.

From the actual tracking perspective, the selection of the feature information must satisfy the following requirements:

1. Independence between features: If the two features is very similar, then select any features or a combination of these two features to build the cost function will not have a huge impact to the tracking results. When selecting the features, the unnecessary feature information must be removed as much as possible in order to reduce the redundancy of information.

2. Complementarity between features: The various selected features must ensure that the type of different features can be applied in different circumstances, and when estimating the human body pose, various features should provide relative complementary statistical information.

3. Features' effectiveness and simplicity: Although some features are proven to be effective, for actual tracking, their computational costs are too large to meet the actual demand. Therefore, simplicity for real-time implementation is as useful as effectiveness.

Therefore, using a variety of features together to complete the tracking task is a meaningful scientific exploration as well as how to conduct feature integration. In this paper, the robust tracking results are obtained by including seven features, silhouette, edge, motion, skin color, and arm silhouette, ratio area, and scale factor.

3.4.1 Silhouette

Silhouette is a widely used feature. Usually, the binary image silhouette is obtained by firstly learning background model from the image sequence, then separating foreground and background using background subtraction algorithm. Almost all tracking based human pose estimation starts from segmenting human body from the image sequence; and then in the

tracking process, the accuracy of tracking algorithms will be largely depending on this feature. Therefore, human body segmentation is a very important steps in pose estimation field.

Background subtraction algorithm is a commonly used method to extract the foreground region from the sequence of video images. Its main idea is to use the difference of the current image and the background image to extract the foreground region. Foreground region is the current frame image by subtracting the background image and the difference image is greater than a certain threshold region. Simple subtraction will have a good performance in general case. It is able to provide the most complete feature information, but in a situation like the foreground is very similar to background, or dynamic scene change, or noisy scene, or the presence of shadow, this algorithm will not be capable to use. For above cases, more complex background modeling methods need to be considered.

Many researchers are committed to developing a different background model in order to reduce the segmentation accuracy influence from lighting, shadows, and scene changing. There are many existing background modeling methods. For example, Gaussian mixture model [56], the main idea is to represent the gradation of each pixel with a plurality of mixing weighted Gaussian distribution, to use statistical methods to build the background model, and to dynamically update in the process.

In our system, since we are doing front-view upper body human pose estimation, people do not always keep moving. Thus, Gaussian mixture model is not a good choice because of the dynamically updating. However, in order to obtain an accurate silhouette feature, we are not just using background subtraction algorithm. Equation 3.4.1 - 1 shows how to obtain the silhouette, where I_{sub} is the result of background subtraction algorithm, and I_{Edge} and I_{skin} will be demonstrated in next sections. Figure 3.4.1 - 1 shows an example.

$$I_{sil} = I_{sub} \cup I_{Edge} \cup I_{skin} \qquad \text{Equation 3.4.1 - 1}$$

Figure 3.4.1 - 1: Silhouette Example

3.4.2 Edge

Edge refers to the gradation of the image mutated region, is one of the basic visual characteristics of human perception. Inside pixels of a same object in an image tend to have very similar gray values, but pixels of different objects tend to have different gray values; so there are more obvious difference between the gray value of all human body parts and the background. Edge feature is invariant to outside lighting, color or texture changes. It does not require a fixed

background. Thus, edge feature is a robust image feature and many researchers use this feature as a reliable image feature for pose estimation.

The shortcoming of edge feature is that it is only a relatively underlying characteristic, which means that it cannot provide more description of the interested target; thus, using only the edge feature cannot distinguish between the background image and the foreground image. Especially when the background image is complex and vulnerable to the interference of human body, edge will provide serious poor feature information in the tracking process.

In digital image processing field, edge detection algorithm are usually based on edge detection operator, such as Log, Sobel, Laplacian, Canny, and so on. These operators are highly sensitive to noise. Canny operator is the most relatively stable operator for edge detection, and it has been widely used. In addition, researchers often use Gaussian filter to smooth the input image to improve the stability of edge detection.

In this paper, the first order derivative images of the current frame in x and y directions are calculated by using Sobel operator. Then, use two background derivative images to subtract the two resulting first order derivative images separately. Next, compute the gradient magnitude image by adding the resulting subtracted images. At last, threshold the gradient image to get the edge feature image I_{Edge}, as Figure 3.4.1 - 1 shows.

3.4.3 Motion

As section 2.5 described, motion feature has three main purpose. One purpose is the motion part detection. Motion parts that are grouped as TORSO, RIGHT ARM, and LEFT ARM. Motion skin image $I_{MotSkin}$ and motion edge image $I_{MotEdge}$ are obtained from Equation 3.4.3 - 1 and 3.4.3 - 2, where I_{Motion} is the absolute difference between current frame and previous frame. Then, compute the total pixel of $I_{MotCombined}$ that is calculated by Equation 3.4.3 - 3. Take the torso part as an example, if the number of the total pixel of torso body part is greater than half of the perimeter of the torso body model, the motion flag of the torso body is triggered.

$$I_{MotSkin} = I_{Motion} \cap I_{Skin} \qquad \text{Equation 3.4.3 - 1}$$

$$I_{MotEdge} = I_{Motion} \cap I_{Edge} \cap I_{Skin} \qquad \text{Equation 3.4.3 - 2}$$

$$I_{MotCombined} = I_{MotSkin} \cap I_{MotEdge} \qquad \text{Equation 3.4.3 - 3}$$

Therefore, if a motion flag of a certain body part is not triggered, our system will not run pose estimation on this certain body part in current frame and keep the previous frame pose data; so that this way can save a lot tracking time. An example of I_{Motion} is showed in Figure 2.5 - 4.

3.4.4 Skin

3.4.4.1 Skin Feature Definition

As section 2.5 described, skin color I_{Skin} is defined within a range of R-G color space. In this paper, in order to obtain a more clean skin color feature image, Equation 3.4.4.1 -1 is used, where I_{Skin0} is the original skin pixel image and I_{BSkin} is the background skin image.

$$I_{Skin} = I_{Skin0} \cap (I_{BSkin} \cup I_{Motion} \cup I_{Sub}) \qquad \text{Equation 3.4.4 -1}$$

3.4.4.2 Skin Blob Labeling

Our system tries to label skin blobs from the obtained skin feature I_{Skin}. For each skin blob S_b, the blob gradient probability P_{SB} is computed based on Equation 3.4.4.2 -1, where $N(S_b)$ is a normalization term and calculated by the number of pixels in S_b. Blobs with low P_{SB} are eliminated.

$$P_{SB}(S_b) = \frac{\sum_{(x_i,\ y_i) \in S_b} I_{Edge}(x_i,\ y_i)}{N(S_b)} \qquad \text{Equation 3.4.4.2 -1}$$

After labeling skin blobs, our system can accurately obtain the position of two hand joints, which will be a very useful indicator for skin color part of our designed cost function that will be introduced later in this 3.4 section.

3.4.5 Enhanced Arm Tracking Silhouette

Since our system is dealing with the front-view upper human body pose estimation, the great challenge is the movement of arms. This is also because human head and torso are always rarely moving in upper body only video source. From later discussion in this 3.4 section, since we estimate human body parts by parts, it is possible to obtain a false minimum cost function by having a wrong region of interest (ROI). For example, Figure 3.4.5 – 1 shows a bad example with a blue rectangle circling the ROI_{bad}; and Figure 3.4.5 – 2 shows a good example with another rectangle ROI_{good}. Obviously, ROI_{good} is bigger than ROI_{bad}. This fact results that in a correct pose estimation situation, the cost value of this body parts in ROI_{good} is larger than the cost value in ROI_{bad}, since there are more pixels in this larger area ROI_{good}. This fact will provide wrong result pose estimation, and this is why we are adding this feature into the cost function to prevent the above situation from happening and to enhance the pose estimation result, like the correct result Figure 3.4.5 – 2 shows.

Since our system tracks the human body parts by part, torso first, right arm and left arm, this enhanced arm silhouette feature is constructed by getting the shoulder joints first, and then using them to obtain outside body ROIs. The example of this arm silhouette feature is shown in Figure 3.4.5 – 3.

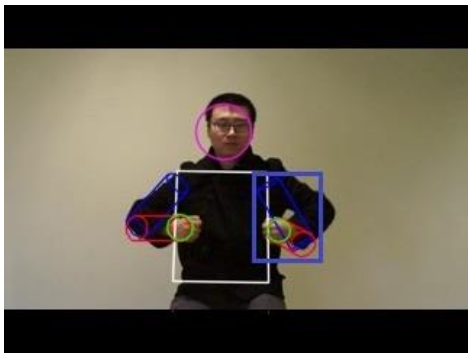


Figure 3.4.5 – 1: Bad Example

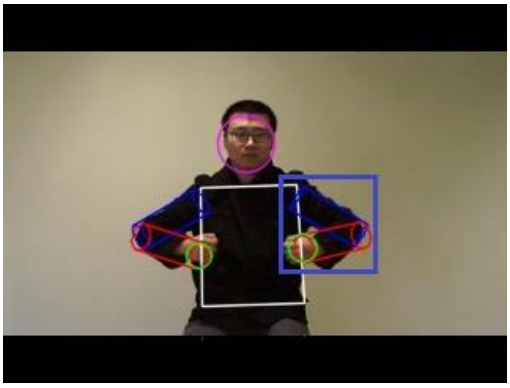


Figure 3.4.5 – 2: Good Example

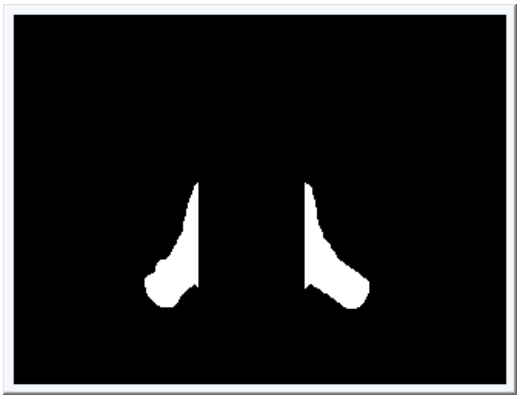


Figure 3.4.5 – 3: Example of Feature Enhanced Arm Tracking Silhouette

3.4.6 Ratio Constraint Silhouette

This feature is inspired by [59]. The function of this term is attracting the silhouette, which means that pushing the projected model image inside the image silhouette. In other words, this term enforces the projected model image $I_{projectedSil}$ to remain within the image silhouette, but also demands that the image silhouette is entirely explained by combining other features. This feature combined with other features in this paper mean that all silhouette parts contribute to the cost function that drives the fitting process. This part of cost function is constructed by calculating the absolute difference between one and the ratio value of the area of projected silhouette divides to the area of image silhouette in a certain blob area S_b, as shown in Equation 3.4.6 – 1.

$$C_{Ratio} = abs(\frac{\sum_{(x_i,\ y_i) \in S_b} I_{projectedSil}(x_i,\ y_i)}{\sum_{(x_i,\ y_i) \in S_b} I_{Sil}(x_i,\ y_i)} - 1) \qquad \text{Equation 3.4.6 – 1}$$

3.4.7 Scale

The scale change is detected when abs($Area_{projected3D} - Area_{Sil}$) is greater than a default threshold, where $Area_{projected3D}$ is the area of projected 3D human body model image with previous estimated pose parameters and $Area_{Sil}$ is the area of silhouette image. After the scale change is detected, the scale of our human body model will be changed based on the detected scale value. Figure 3.4.7 – 1 shows an example of scale changing, and this figure is from [1] since we are using similar system designs.

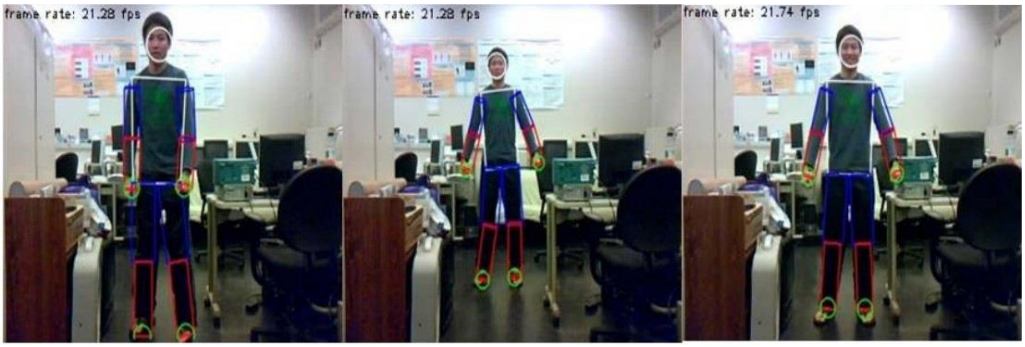


Figure 3.4.7 – 1: Example of Scale Change

From previous sections, the projected image from 3D human body model is calculated by Euler's angle rotation and transformation matrices. However, in order to obtain a correct coordinate for 3D pose parameters, our system needs a world to camera transformation matrix and a scale matrix to adjust the detected scale properly. Matrix Figure 3.4.7 – 2 shows the world to camera transformation matrix, and matrix Figure 3.4.7 – 3 shows the scale centroid matrix; where scale is the detected scale and (cx, cy) is the calculated centroid from silhouette image.

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$
Figure 3.4.7 – 2: World to Camera Transformation

$$\begin{pmatrix} scale & 0 & 0 & cx \\ 0 & scale & 0 & cy \\ 0 & 0 & scale & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$
Figure 3.4.7 – 3: Scale and Centroid Matrix

3.4.8 Image Cost Value and Cost Function

Equation 3.4.8 – 1 shows an example of image matching cost function. The current upper and lower arm length is calculated first; and next, the image costs are computed in a valid search

region sampled with uniform step for all possible body configurations. Then choose the maximum cost from all image costs through our iterative APSO algorithm. In Equation 3.4.8 – 1, $S_b(i, j, l)$ means one possible 2D body part configuration, where l is the iterative number from all possible body part configurations; length($S_b(i, j, l)$) means the body part configuration's length. Figure 3.4.8 – 1 shows a graph example of constructing this image matching cost function.

$$C_{Sil}(i,j) = \max_l\left(\frac{\sum_{(x_n,\ y_n)\in S_b(i,j,l)} I_{Xor}(x_n,\ y_n)}{length(S_b(i,j,l))}\right) \qquad \text{Equation 3.4.8 – 1}$$

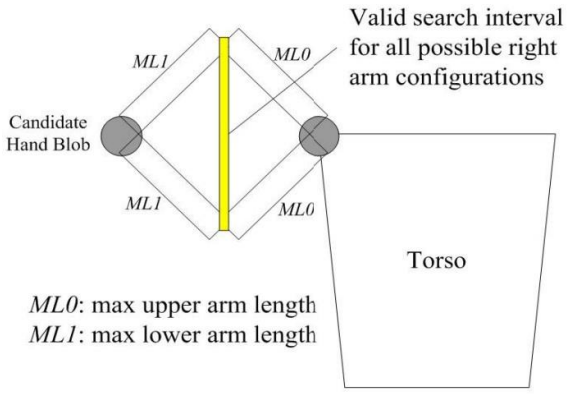


Figure 3.4.8 – 1: Graph Example of Image Matching Cost Function

3D human upper body pose is estimated for the best matching image of torso and two arms by using the APSO algorithm to minimize the cost function $Cost_{image\_model}$, as shown in Equation 3.4.8 – 2, where it contains silhouette score, edge score, motion score, skin points score, enhanced arm score, and ratio score; and $w_0 = 3$, $w_1 = 20$, $w_2 = 15$, $w_3 = 250$, $w_4 = 4$, and $w_5 = 400$.

$$Cost_{image\_model} = w_0 * Score_{Sil} - w_1 * Score_{edge} - w_2 * Score_{motion} + w_3 * Score_{skin} + w_4 * Score_{arm} + w_5 * Score_{ratio}$$

Equation 3.4.8 – 2

Take an example for silhouette score, the variable I_{Xor} used in Equation 3.4.8 – 1 comes from Equation 3.4.8 – 3. The similar computational method happens to edge score from Equation 3.4.8 – 4, motion score from Equation 3.4.8 – 5, and enhanced arm score from Equation 3.4.8 – 6.

$$I_{silXor} = N_{XOR}(I_{silImage}, I_{projectedModelSil}) \qquad \text{Equation 3.4.8 – 3}$$

$$I_{edgeAnd} = N_{AND}(I_{edgeImage}, I_{projectedModelEdge}) \qquad \text{Equation 3.4.8 – 4}$$

$$I_{motionAnd} = N_{AND}(I_{motionImage}, I_{projectedModelEdge}) \qquad \text{Equation 3.4.8 – 5}$$

$$I_{armXor} = N_{XOR}(I_{silImage}, I_{projectedModelSilArm}) \qquad \text{Equation 3.4.8 – 6}$$

For skin points' score, it is measured by computing the square root distance between the joint analyzed from 2D image information P_{image} and the joint estimated from 3D model P_{model}, as shown in Equation 3.4.8 – 7, where i = 1,2,3 represents for head joint, right hand joint, and left hand joint.

$$Score_{skin} = \sum_{i=1}^{3} \sqrt{(P_{image}^{i} - P_{model}^{i})^2} \qquad \text{Equation 3.4.8 – 7}$$

In addition, ratio score is calculated by Equation 3.4.6 – 1. Moreover, all image variables $I_{xxxxxImage}$ have been mentioned from previous sections.

Our best 3D pose is searched in a hierarchical way, which means that taking head and torso as a base followed by two arms, as shown in Figure 3.4.8 – 2.

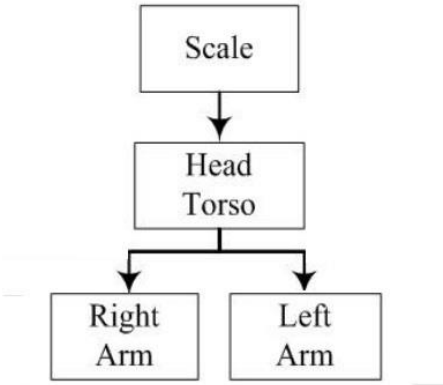


Figure 3.4.8 – 2: Hierarchical Search Example

3.4.9 Tracking Lost Recovery

In our designed system, the mean shift model tracks two individual targets, right arm and left arm. Thus, it is possible to have tracking failure. In order to prevent accumulating tracking errors, a lost detector is triggered if dis($P_{image,}$ P_{model}) (calculated the same way as Equation 3.4.8 – 7 shows) is greater than a default threshold. After confirming tracking lost, instead of using trajectory constraint (introduced in the next section) to redistribute particles, our system goes back to use initial constraint to redistribute particles to recover the tracking lost.

3.5 Human Motion Constraint Module

In the tracking process, due to the structure body itself, the complexity of human movement, the body parts occlusion with each other, and self-occlusion, these facts may result the rapid failure in the tracking process. Therefore, it is necessary to introduce human motion constraints to prevent the accumulation of errors in the tracking process.

The kinesiology constraint refers to human movement should comply with certain restrictions, including the motion range of the joint angle constraint [30][57] and the constraint that various parts of the body cannot penetrate each other. These constraints can be used as hard constraints

to divide the state space into the legitimate and illicit part, in order to reduce the search range; they can also be used as soft constraints, i.e. punishment factor [58] proposes. Paper [58] creates a body mutual penetration cost function in its tracking framework, in order to avoid the interpenetration of the body part. Any human body model has constraints to restrict the model act like real human. The benefit of adding constraints is to greatly reduce the search space; and the disadvantage is that when dealing with these constraints in the optimization process, it increases the amount of computation.

The angle constraint refers to the rotation range of the angle of each joint. Because of the body's own biological characteristic, each person's range of rotation of each joint is limited within a certain scope. According to this characteristic, we can give a constraint to do this limitation, in order to prevent the continuous accumulation of errors in the tracking process and to achieve a more precise tracking of the human body movement.

This paper's constraint serves three purposes: the angle constraint, non-penetrating constraint, and trajectory constraint. The trajectory constraint means that after using the initial constraint to construct particles in the APSO algorithm, the next time redistributing the particles will not be necessary to start from initial constraint. Therefore, we can redistribute these particles based on previous frame estimated pose data. Table 3.5 – 1 shows the actual constraints that are used in this paper, where $Y_{Correct}$, $X_{Correct}$, $Z_{Correct}$, $Y_{upCorrect}$, $X_{upCorrect}$, $Z_{upCorrect}$, $X_{downCorrect}$, and $Y_{downCorrect}$ mean the correct estimated pose data from previous frame estimation, i.e. $Y_{Correct}$ means the previous correct estimated y angle for torso body parts.

Variable belongs to	Variable name	Initial Constraints	Max Angle	Min Angle
X_{Torso}	y	[-PI, PI]	$Y_{Correct}$ + 0.15	$Y_{Correct}$ - 0.15
	x	[-PI, PI]	$X_{Correct}$ + 0.15	$X_{Correct}$ - 0.15
	z	[-PI/4, 0]	$Z_{Correct}$ + 0.1	$Z_{Correct}$ - 0.1
X_{RArm}	y_{up}	[-PI, 0]	$Y_{upCorrect}$ + 0.15	$Y_{upCorrect}$ - 0.15
	x_{up}	[0, PI/3], [PI/1.4, PI]	$X_{upCorrect}$ + 0.15	$X_{upCorrect}$ - 0.15
	z_{up}	[-PI/4, 0]	$Z_{upCorrect}$ + 0.1	$Z_{upCorrect}$ - 0.1
	x_{down}	[0, PI]	$X_{downCorrect}$+0.15	$X_{downCorrect}$-0.15
	y_{down}	[-PI, PI]	$Y_{downCorrect}$ + 0.2	$Y_{downCorrect}$ - 0.2
X_{LArm}	y_{up}	[0, PI]	$Y_{upCorrect}$ + 0.15	$Y_{upCorrect}$ - 0.15
	x_{up}	[0, PI/3], [PI/1.4, PI]	$X_{upCorrect}$ + 0.15	$X_{upCorrect}$ - 0.15
	z_{up}	[-PI/4, 0]	$Z_{upCorrect}$ + 0.1	$Z_{upCorrect}$ - 0.1
	x_{down}	[0, PI]	$X_{downCorrect}$+0.15	$X_{downCorrect}$-0.15
	y_{down}	[-PI, PI]	$Y_{downCorrect}$ + 0.2	$Y_{downCorrect}$ - 0.2

Table 3.5 – 1: Actual Constraints

3.6 Chapter Conclusion

This chapter details the design and implementation of the system framework and the basic idea that the system is based on, 3D human upper body pose estimation using annealing particle

swarm optimization. The system flowchart is shown in Figure 3.6 -1. First, this chapter talks the start and basic idea of particle swarm optimization. Then, it explains how and why annealing particle swarm optimization (APSO) is used for the optimization method. APSO's flowchart is shown in Figure 3.6 – 2. Third, it presents how and which the human body model is constructed. Next, human detection and its preparation are introduced. Image feature extraction and cost function directly affect the accuracy of the predicted evaluation results. This chapter gives the silhouette, edge, motion, skin, arm silhouette, and ratio six image features to build the cost function. Further add several types of constraints to human pose estimation in the tracking process.

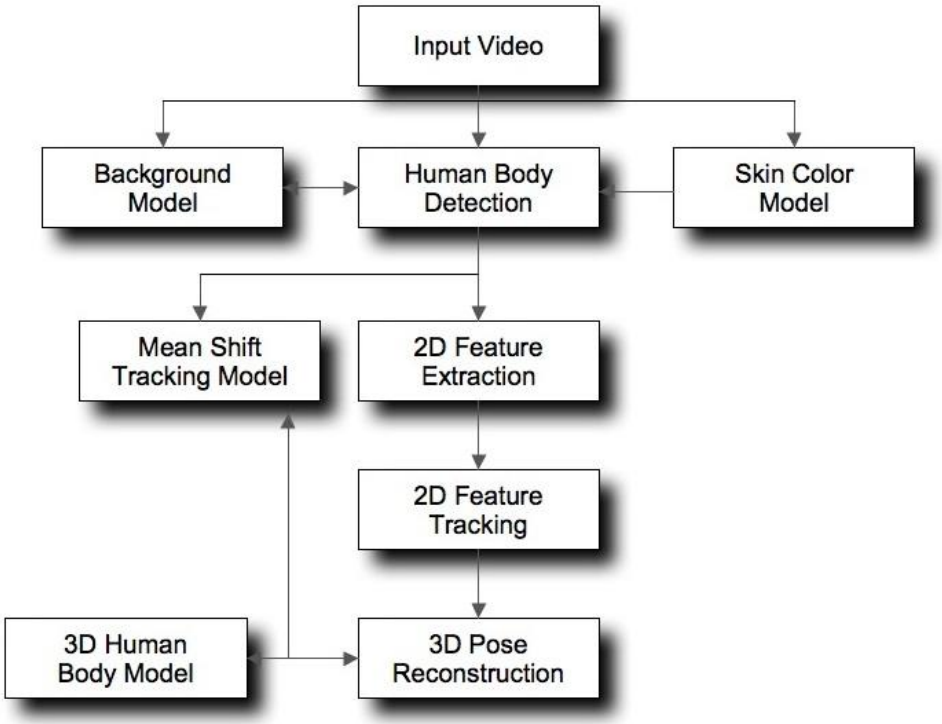


Figure 3.6 -1: System Flowchart

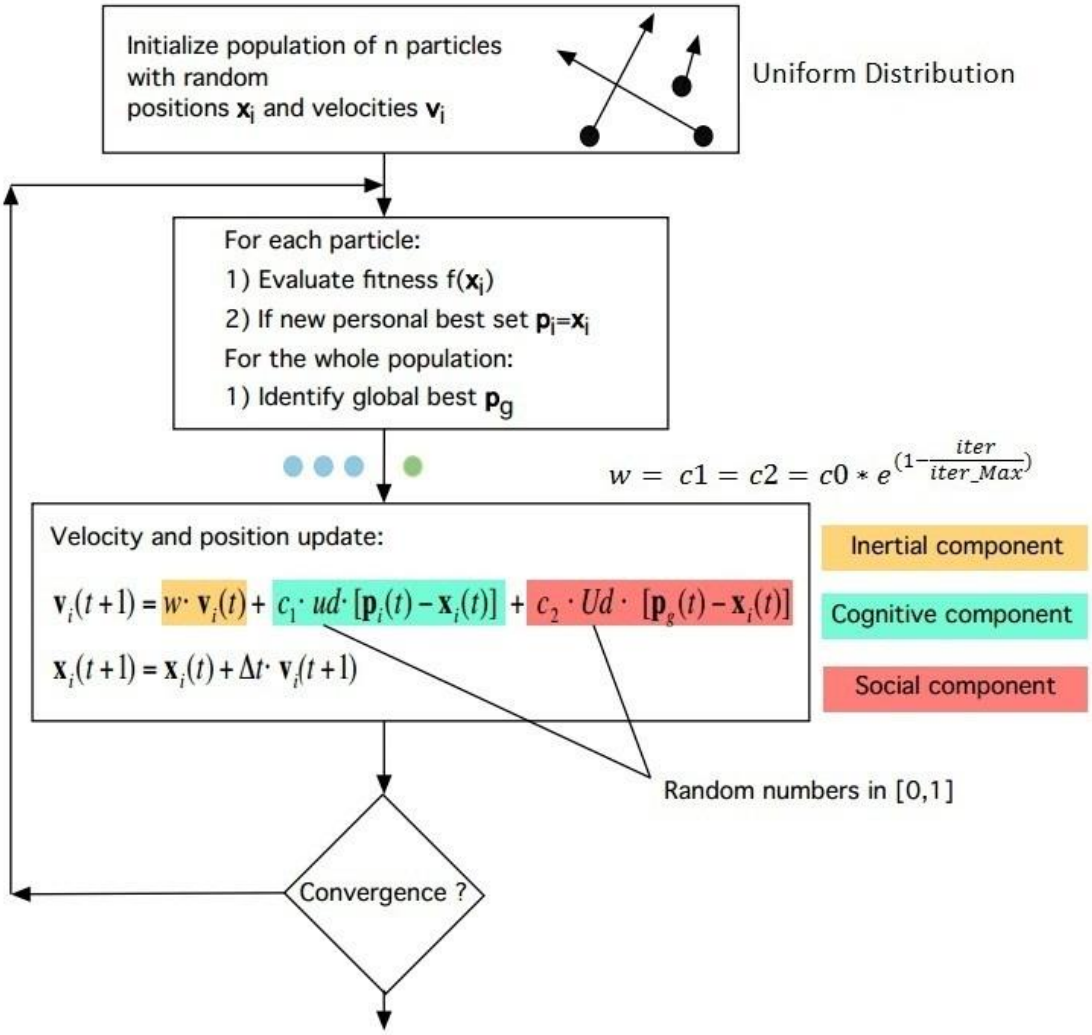


Figure 3.6 -2: APSO Flowchart

Chapter 4:
EXPERIMENTAL RESULT AND ANALYSIS

4.1 Experimental Data

The experimental data is recorded using our own camera device. The reason we are using our own device is that there is no free online front-view upper body pose estimation data available. Therefore, we decide to record video data by ourselves, including testing video data 1 to 7 and ground-truth video data 1 to 4. For ground-truth data, in order to save the trouble of borrowing motion capture equipments, we decide to use Kinect to record our ground-truth video data. Paper [60] shows that Kinect can provide a reasonable accurate result within a close distance, 3 - 4 meters. Upper human body pose estimation requires a close distance between camera and test subject, since one of target applications is to estimate human random pose when they are relaxing on their home sofa. Thus, Kinect ground-truth video data are valid.

Kinect is a Microsoft product that is introduced in section 1.2.1. The default RGB video stream uses 8-bit VGA resolution (640 × 480 pixels) with a Bayer color filter, but the hardware is capable of resolutions up to 1280x1024 (at a lower frame rate) and other color formats. The depth sensing video stream from its IR camera is in VGA resolution (640 × 480 pixels) with 11-bit depth, which provides 2,048 levels of sensitivity.

Test datasets are recorded by using Sony NEX-5 camera. This camera sensor is 23.4 × 15.6 mm EXMOR APS-C HD CMOS Sensor. It has the capability to shoot 1920x1080i at 60 frame/s in AVCHD or 1440x1080p at 30 frame/s in MPEG4, which in this case is the 30 frame/s in MPEG4. Test datasets include different person with different clothes under different backgrounds. Various datasets, including many possible typical human postures such as eating or drinking, provide the typicality for pose estimation.

The experimental data is analyzed under University of Washington lab computer desktop, CPU Intel Core i5-2320 @ 3.00GHz, RAM 6.00 GB, 64-bit Operating System, Windows 7; and the entire system is implemented in C++ using Microsoft Visual Studio 2008, including sub-libraries Open-CV and Open-GL.

4.2 Experimental Result

Figure 4.2 - 1 shows image snapshots examples of ground-truth datasets 1 to 4 from Kinect.

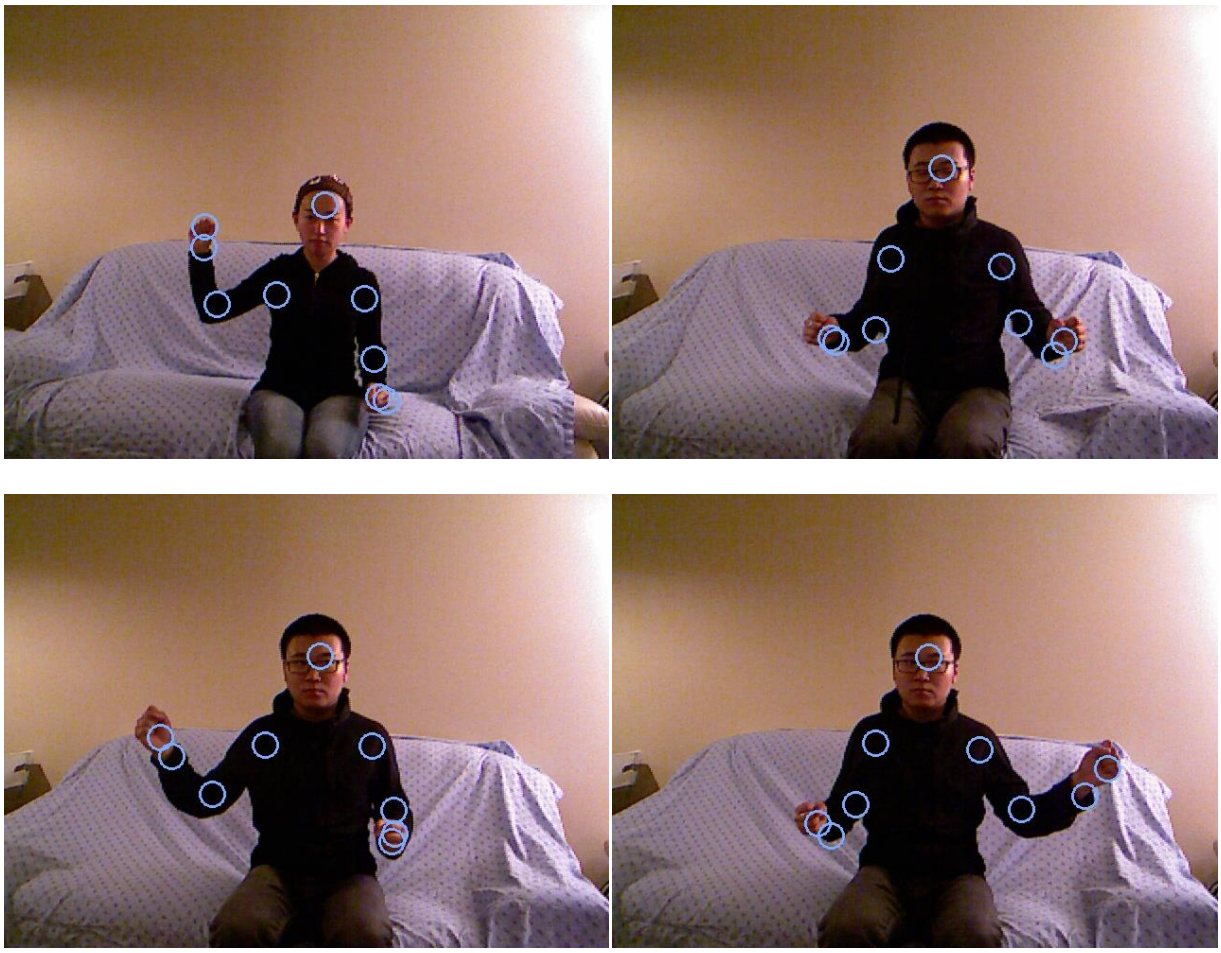
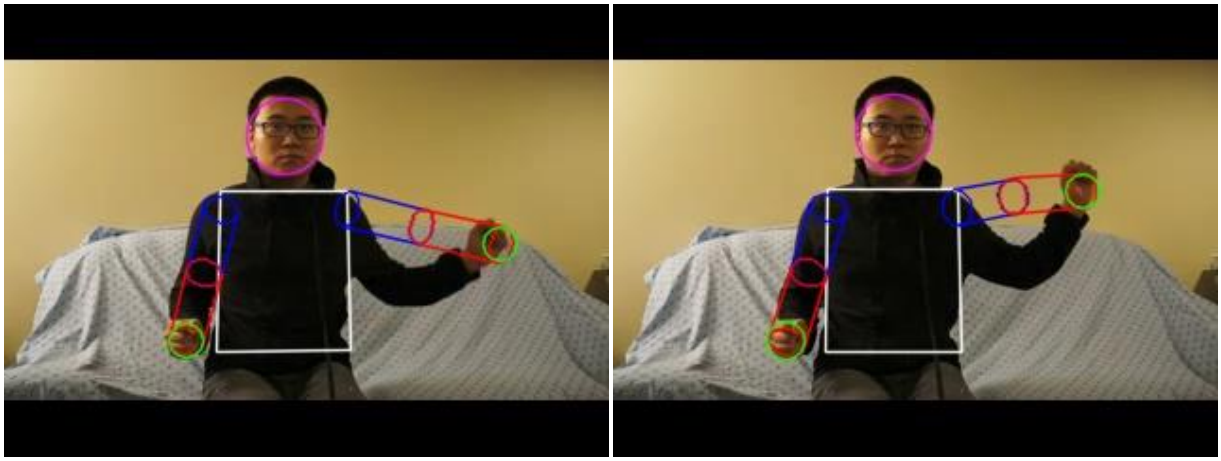


Figure 4.2 - 1: Ground-truth Examples

After implementation of paper [1], Figure 4.2 - 2 shows image snapshot result examples using ground-truth dataset.



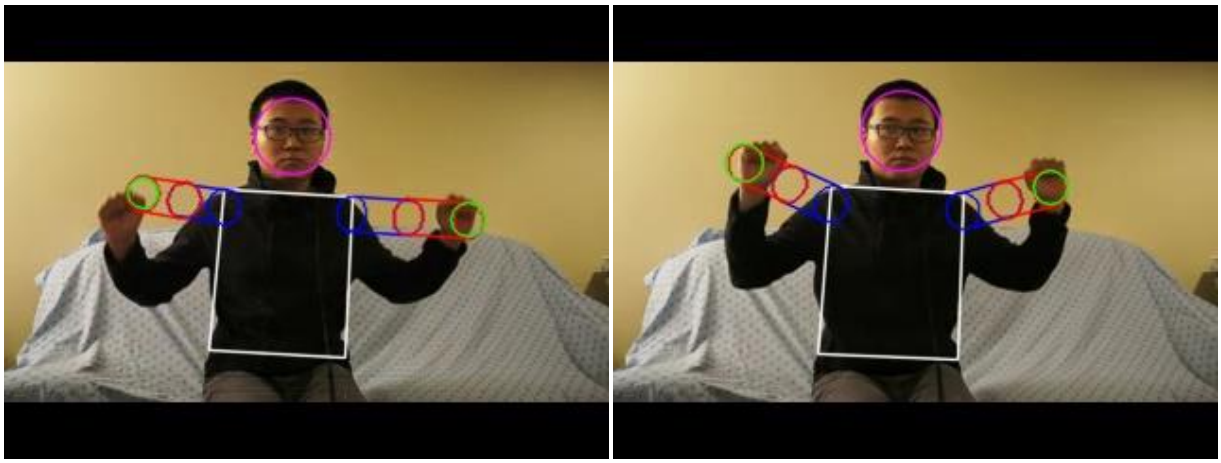
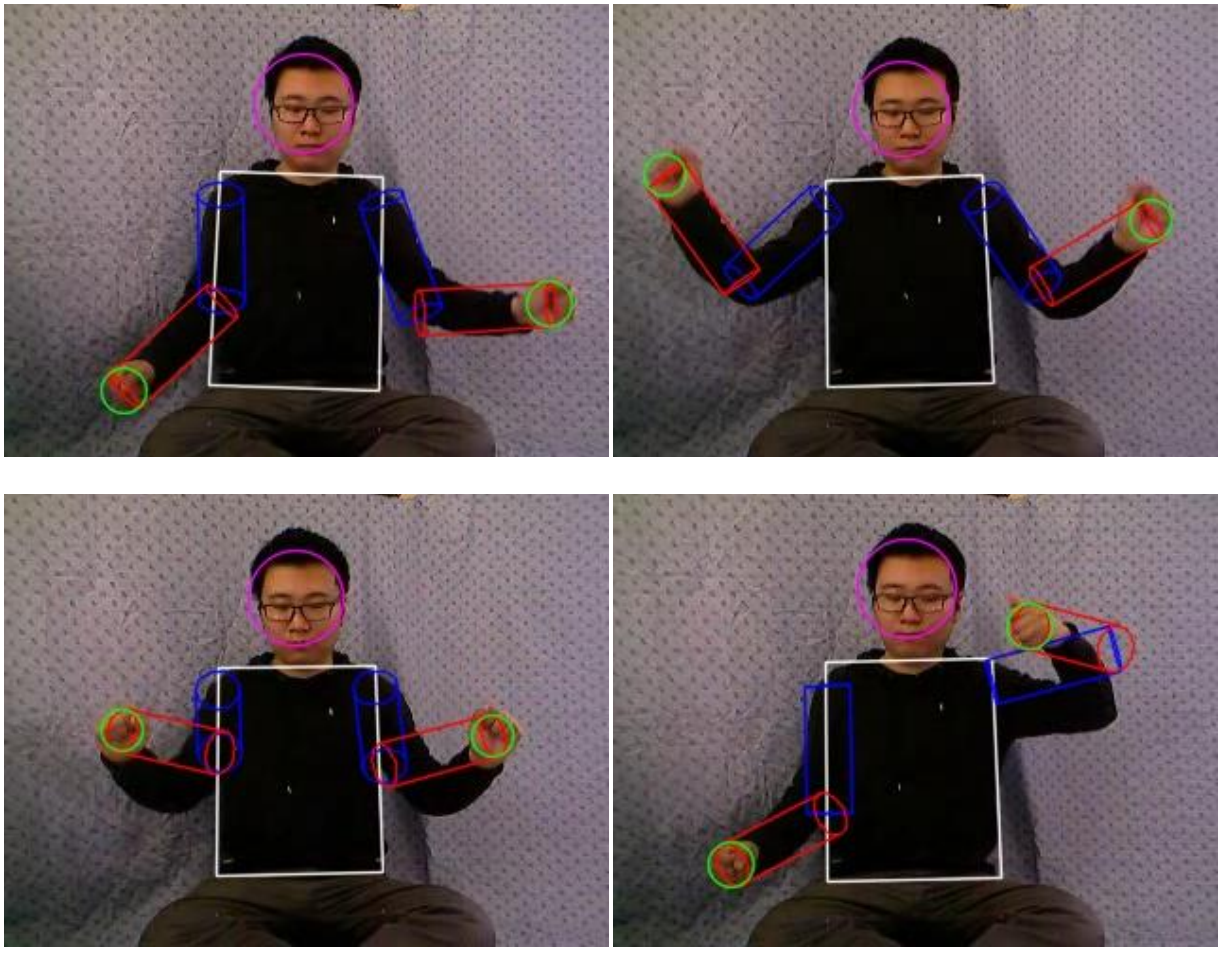


Figure 4.2 - 2: [1]'s Example

After implementation of paper [2], Figure 4.2 - 3 shows image snapshot result examples using ground-truth dataset.



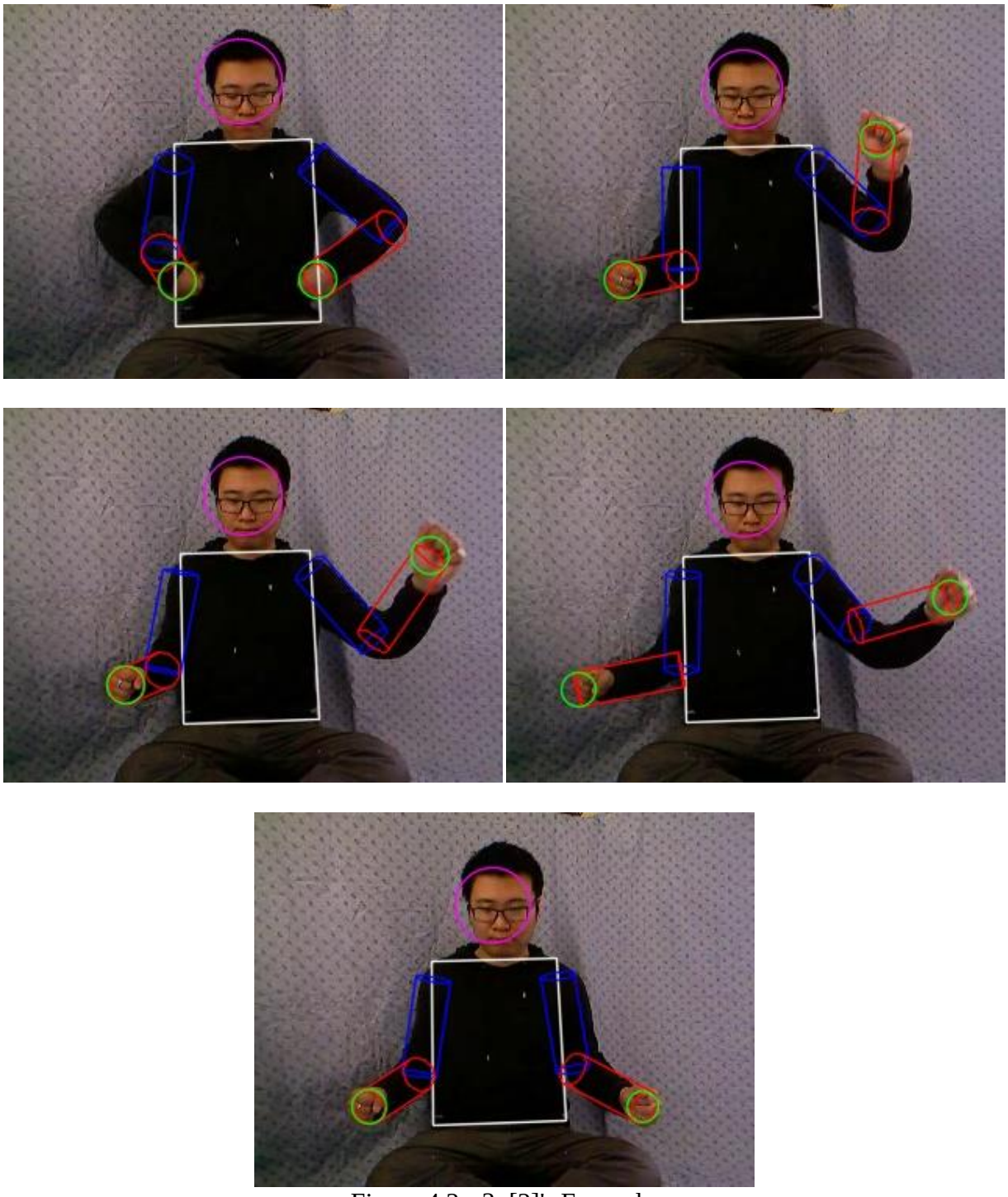
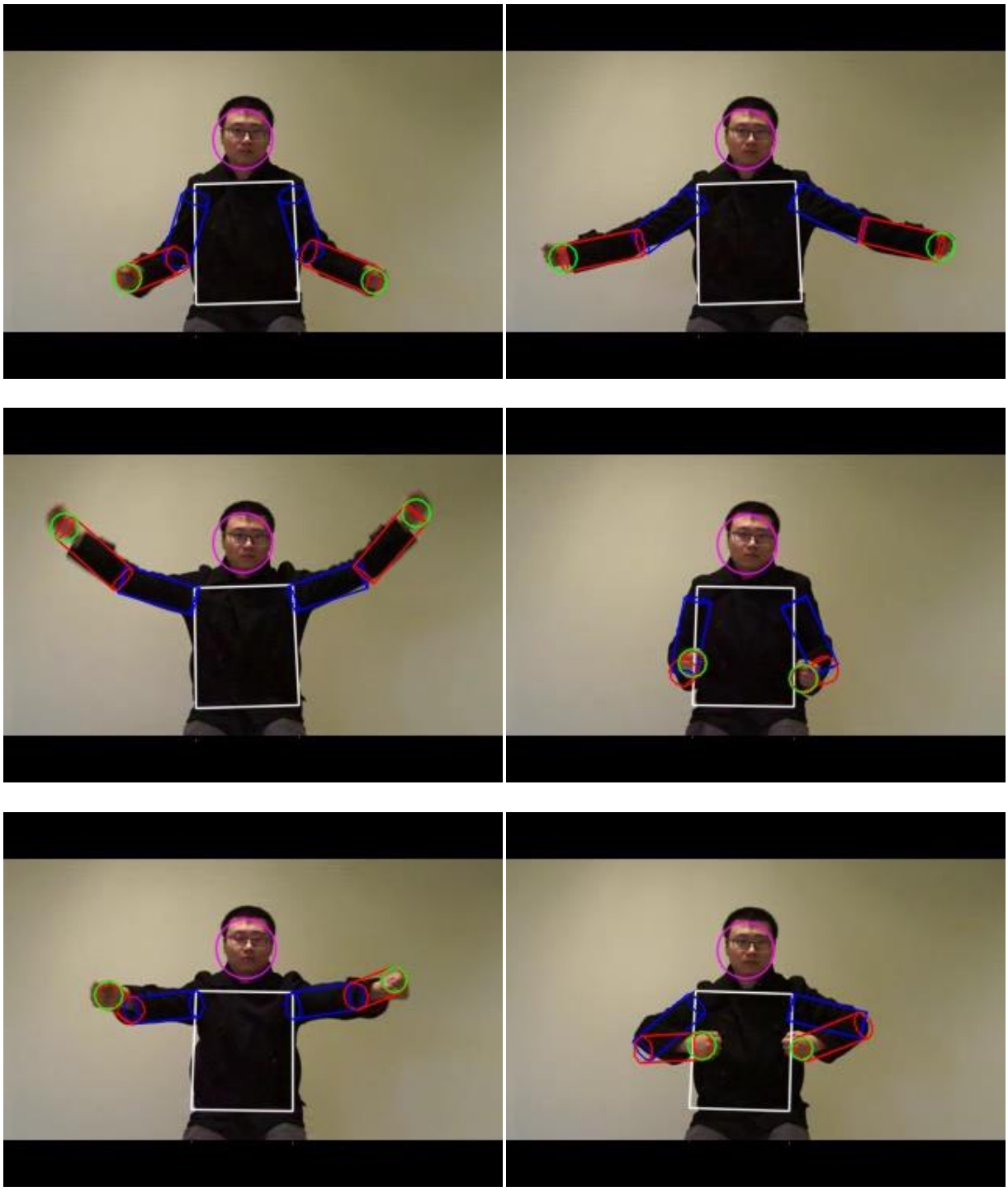
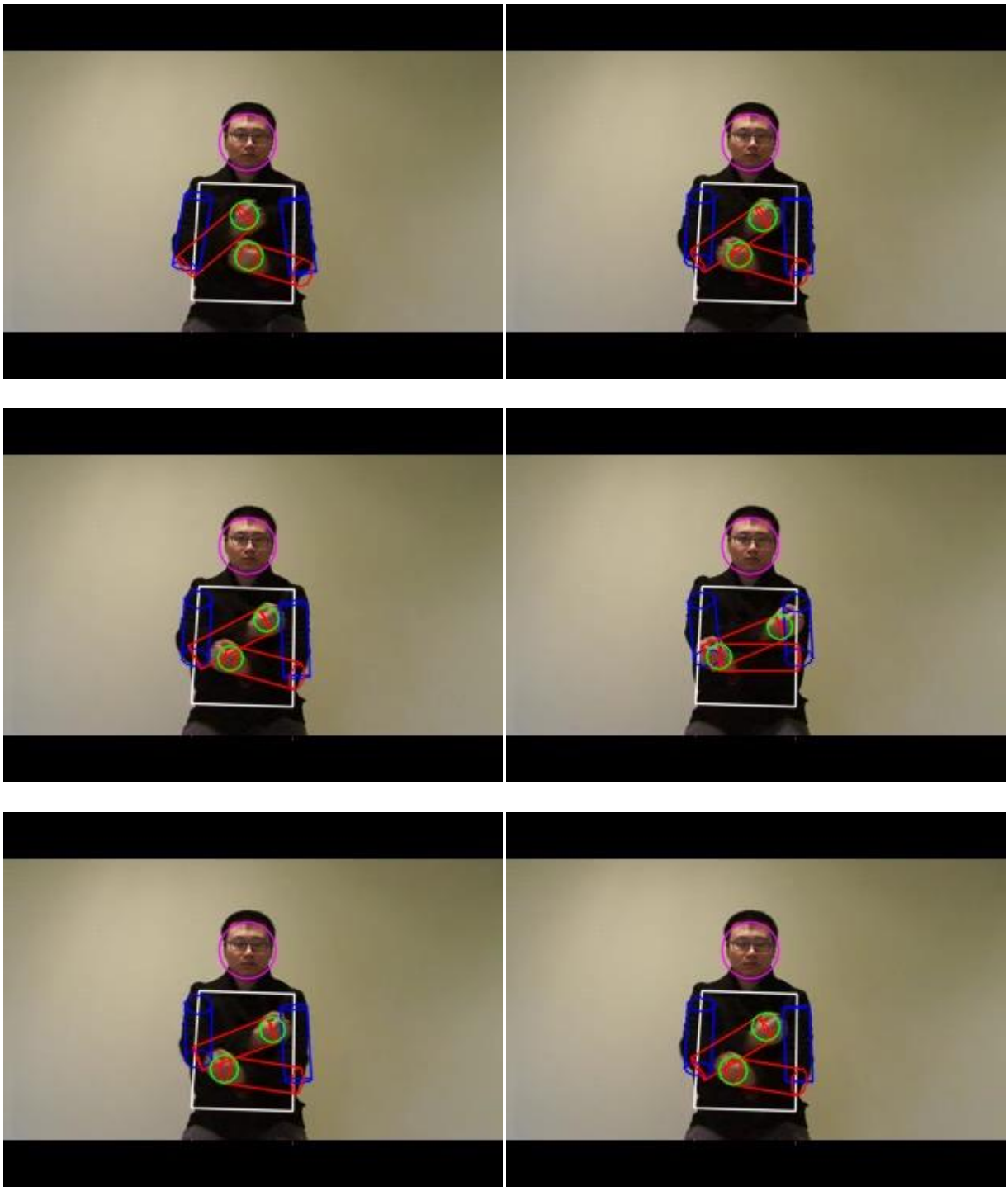


Figure 4.2 - 3: [2]'s Example

From our system implementation, the following Figures show image snapshot result examples using ground-truth datasets and video datasets.

Figure 4.2 - 4 shows video dataset-1 results.





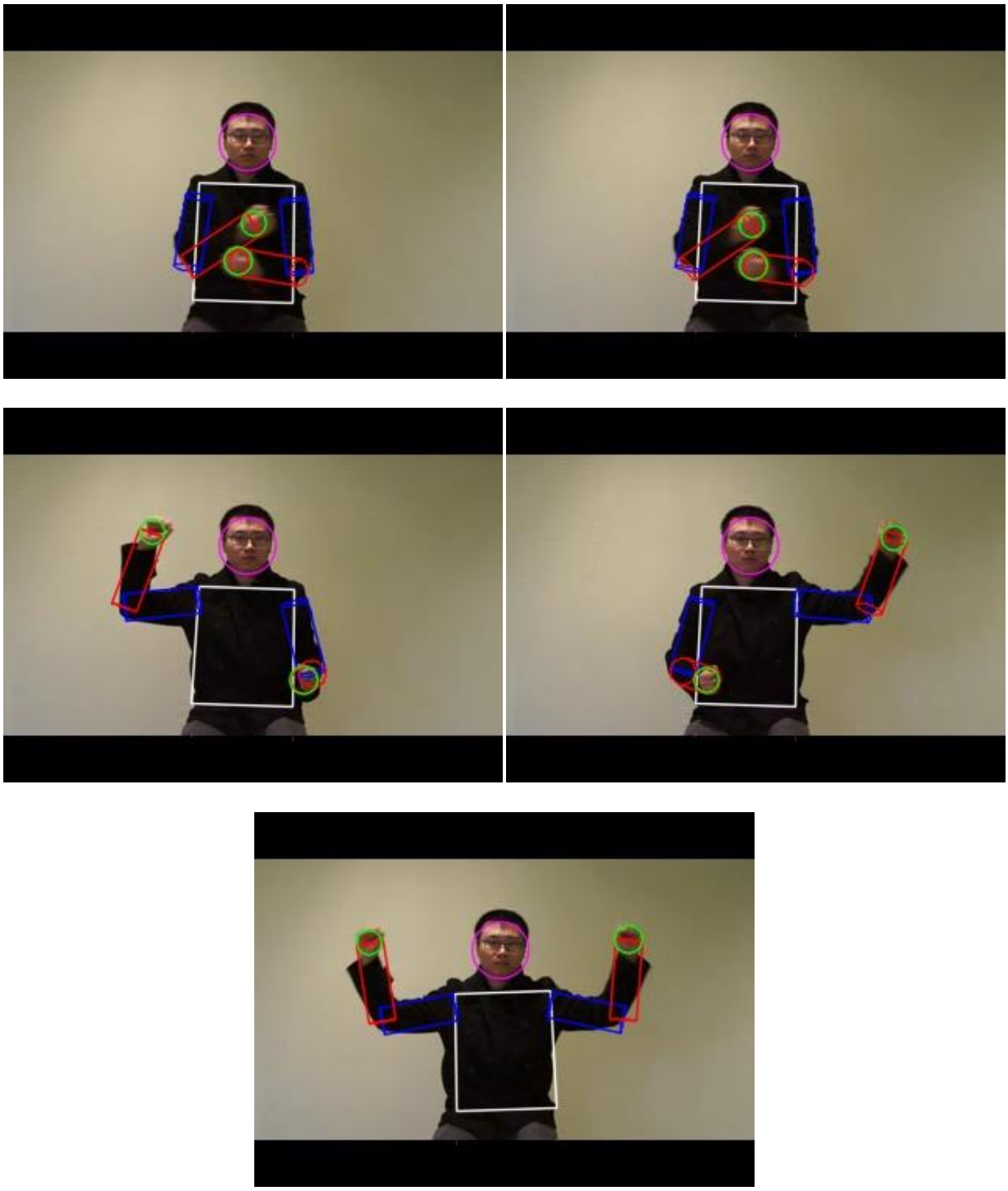


Figure 4.2 - 4: Dataset 1 Results

Figure 4.2 - 5 shows video dataset-2 results.

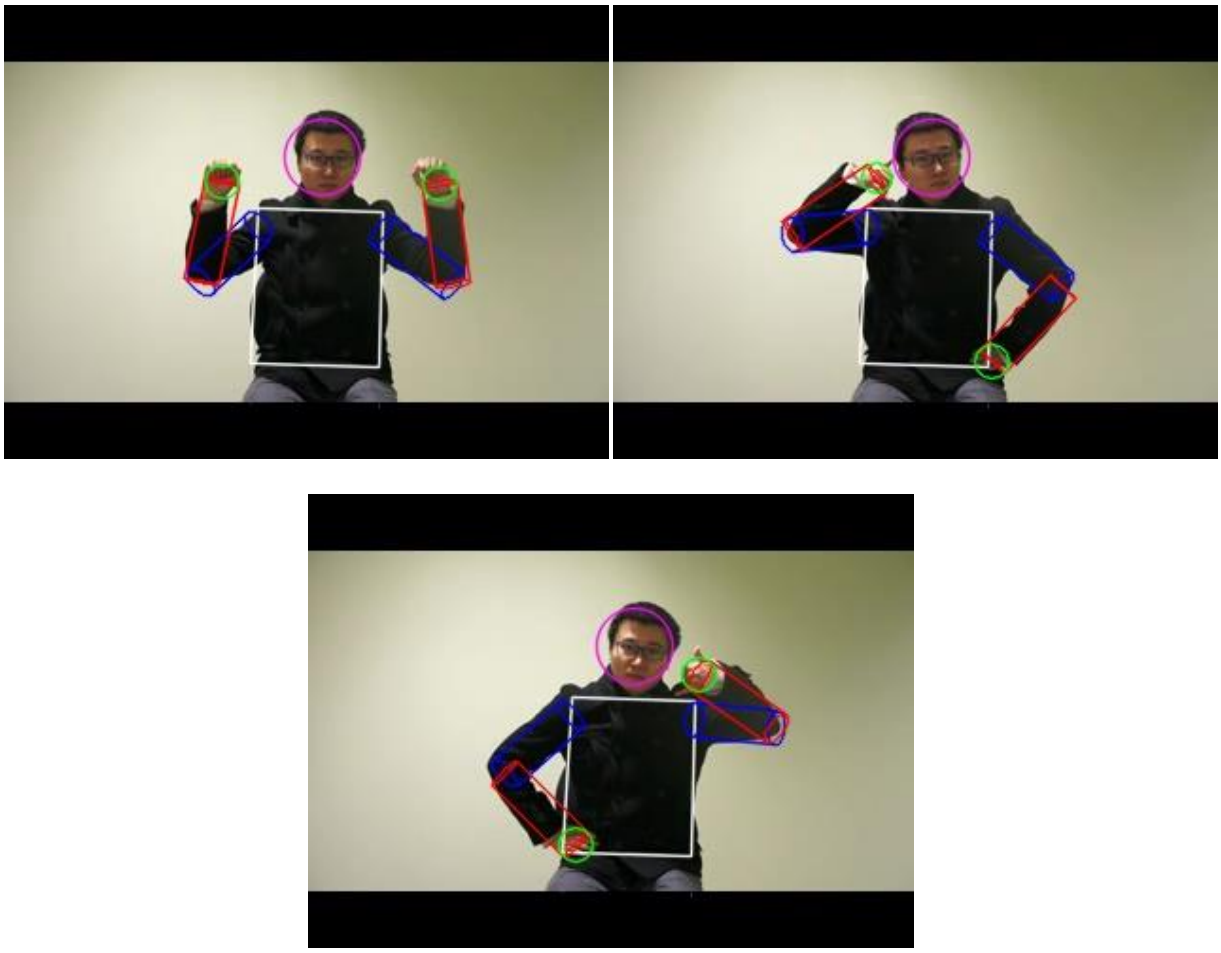
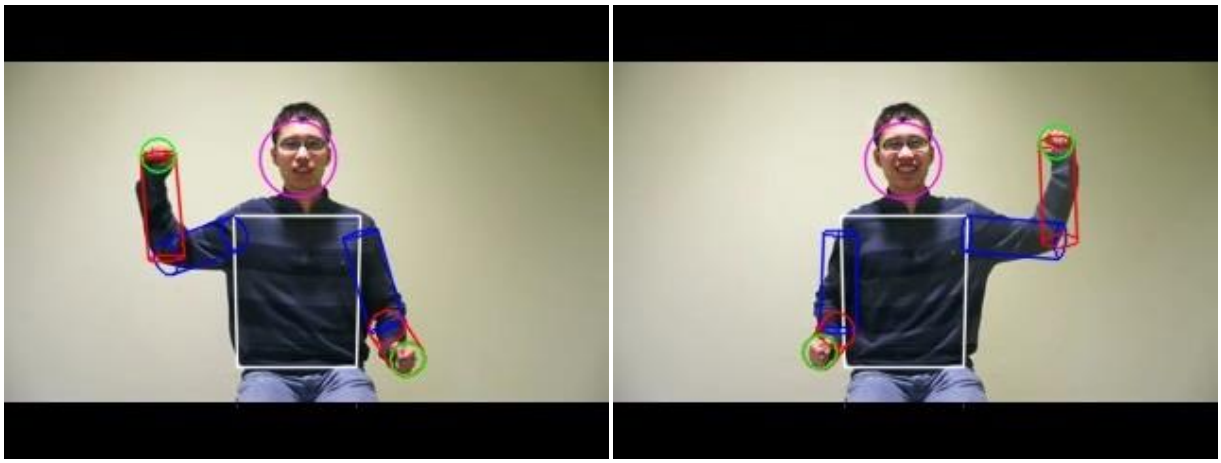


Figure 4.2 - 5: Dataset 2 Results

Figure 4.2 - 6 shows video dataset-3 results.



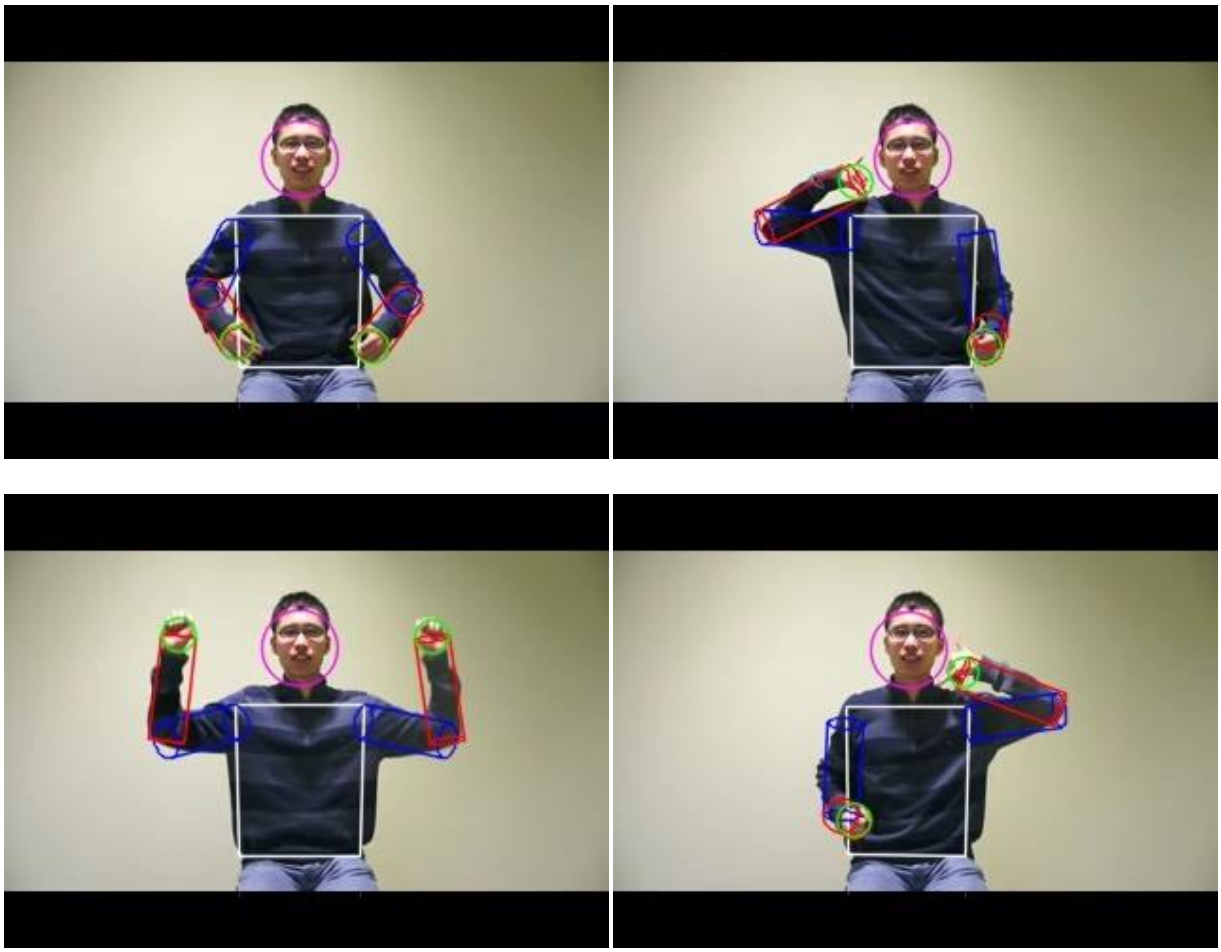
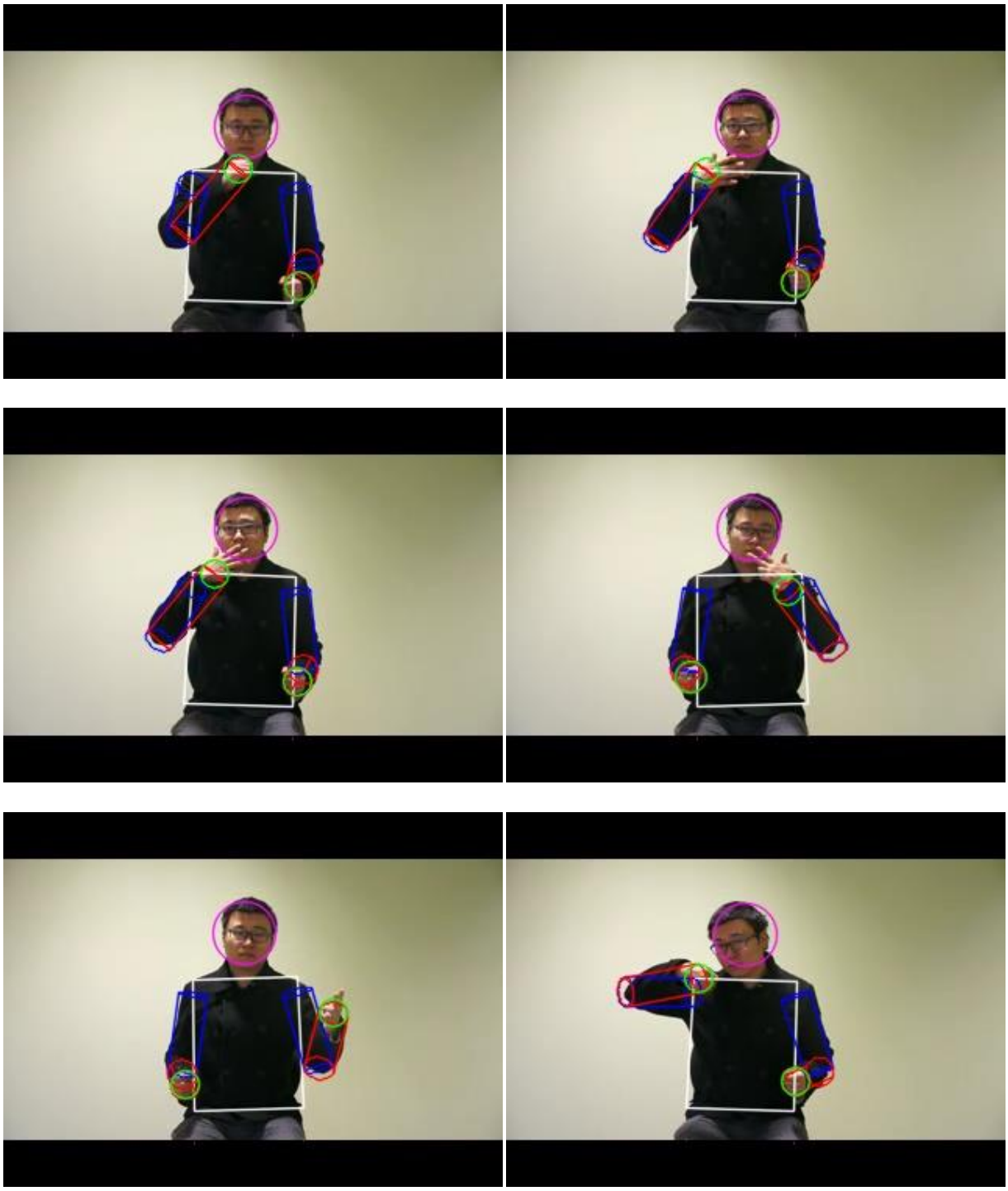


Figure 4.2 - 6: Dataset 3 Results

Figure 4.2 - 7 shows video dataset-4 results.



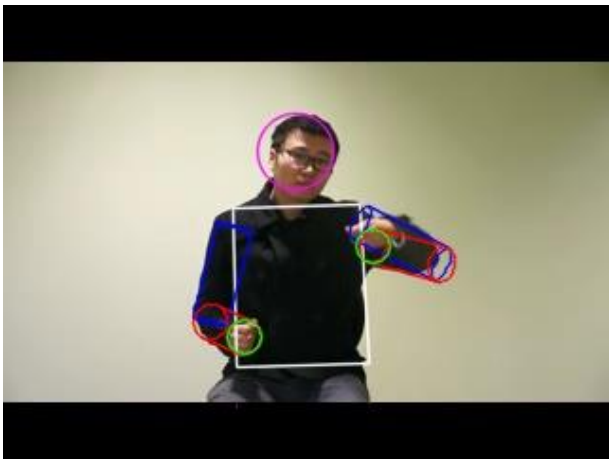


Figure 4.2 - 7: Dataset 4 Results

Figure 4.2 - 8 shows video dataset-5 results.

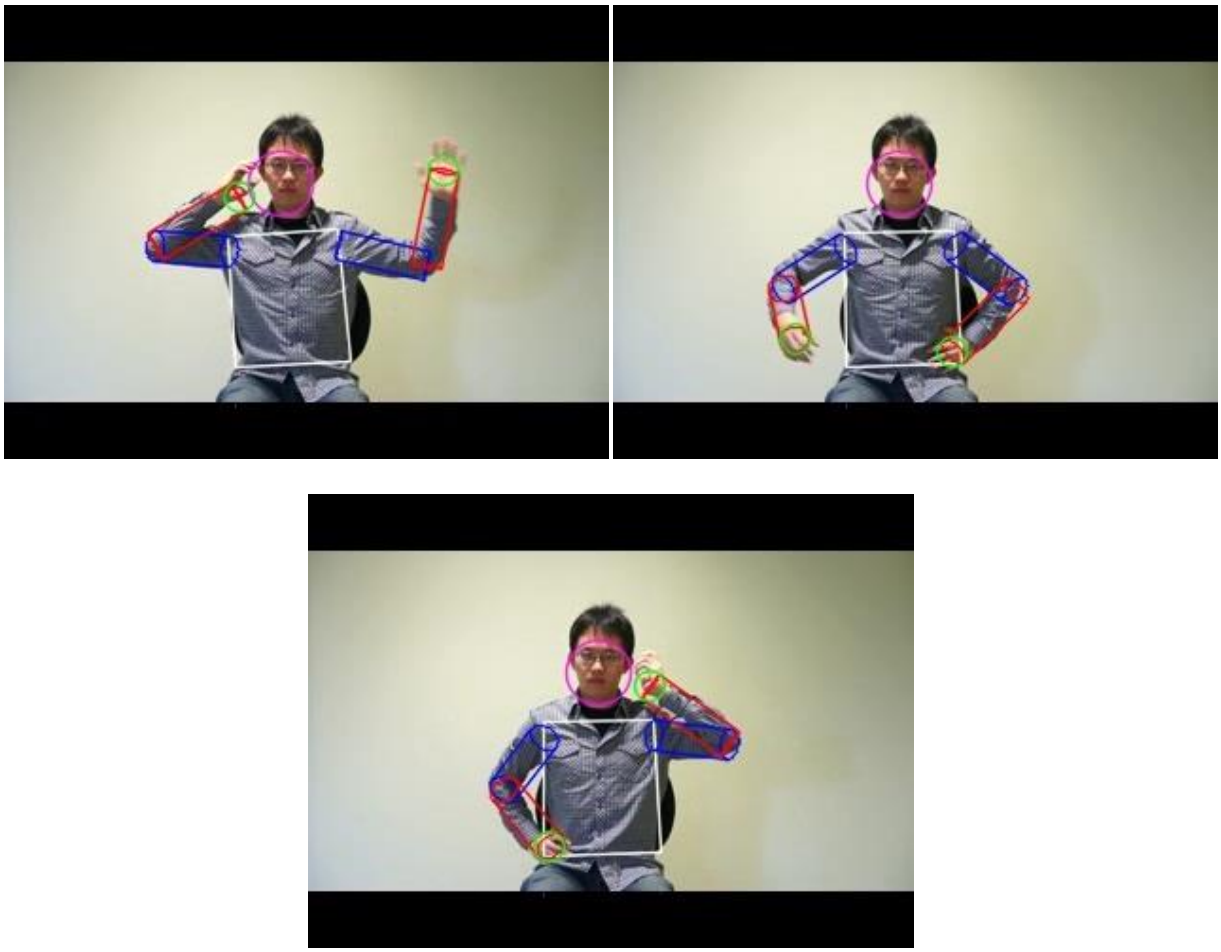
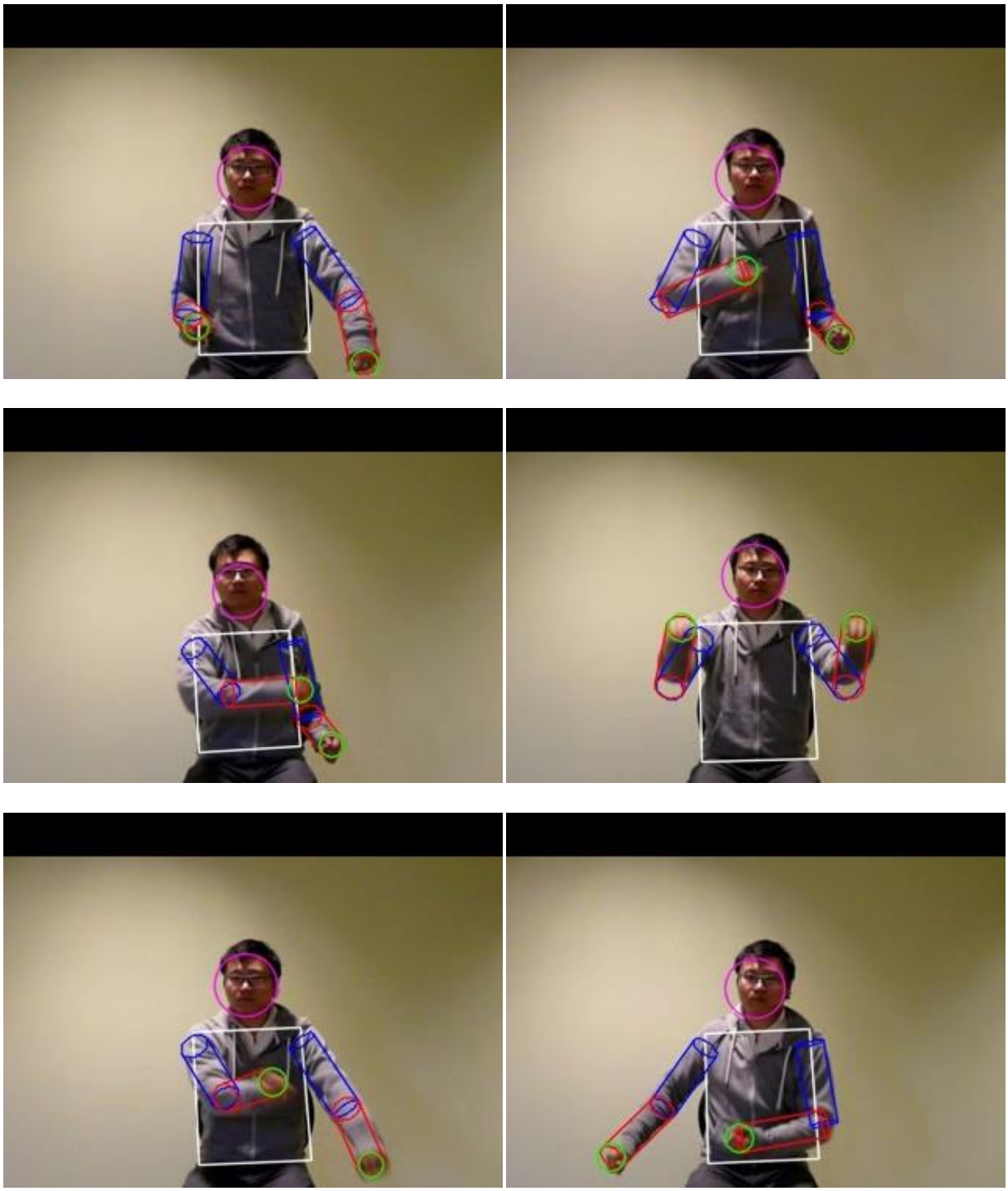


Figure 4.2 - 8: Dataset 5 Results

Figure 4.2 - 9 shows video dataset-6 results.



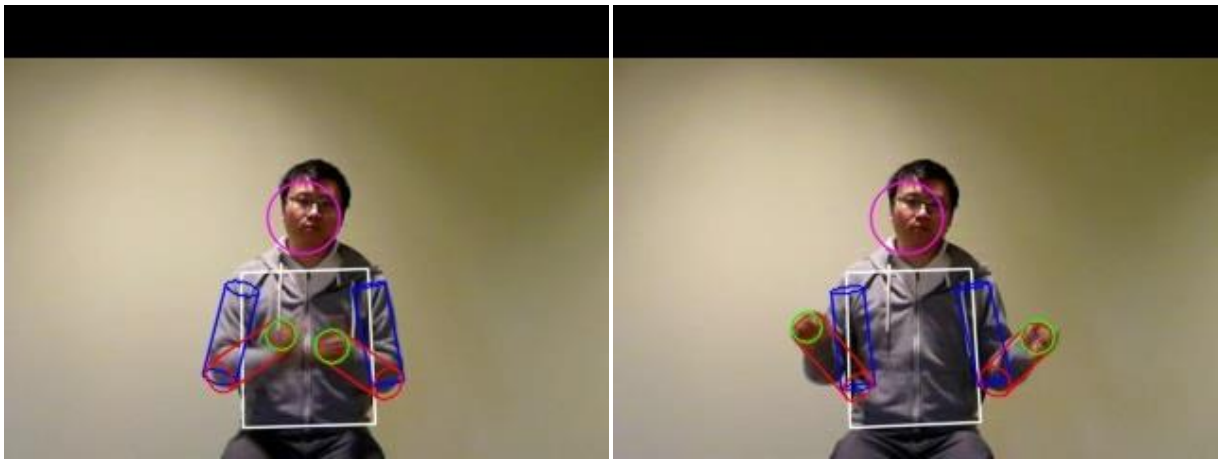


Figure 4.2 - 9: Dataset 6 Results

Figure 4.2 - 10 shows video dataset-7 results.

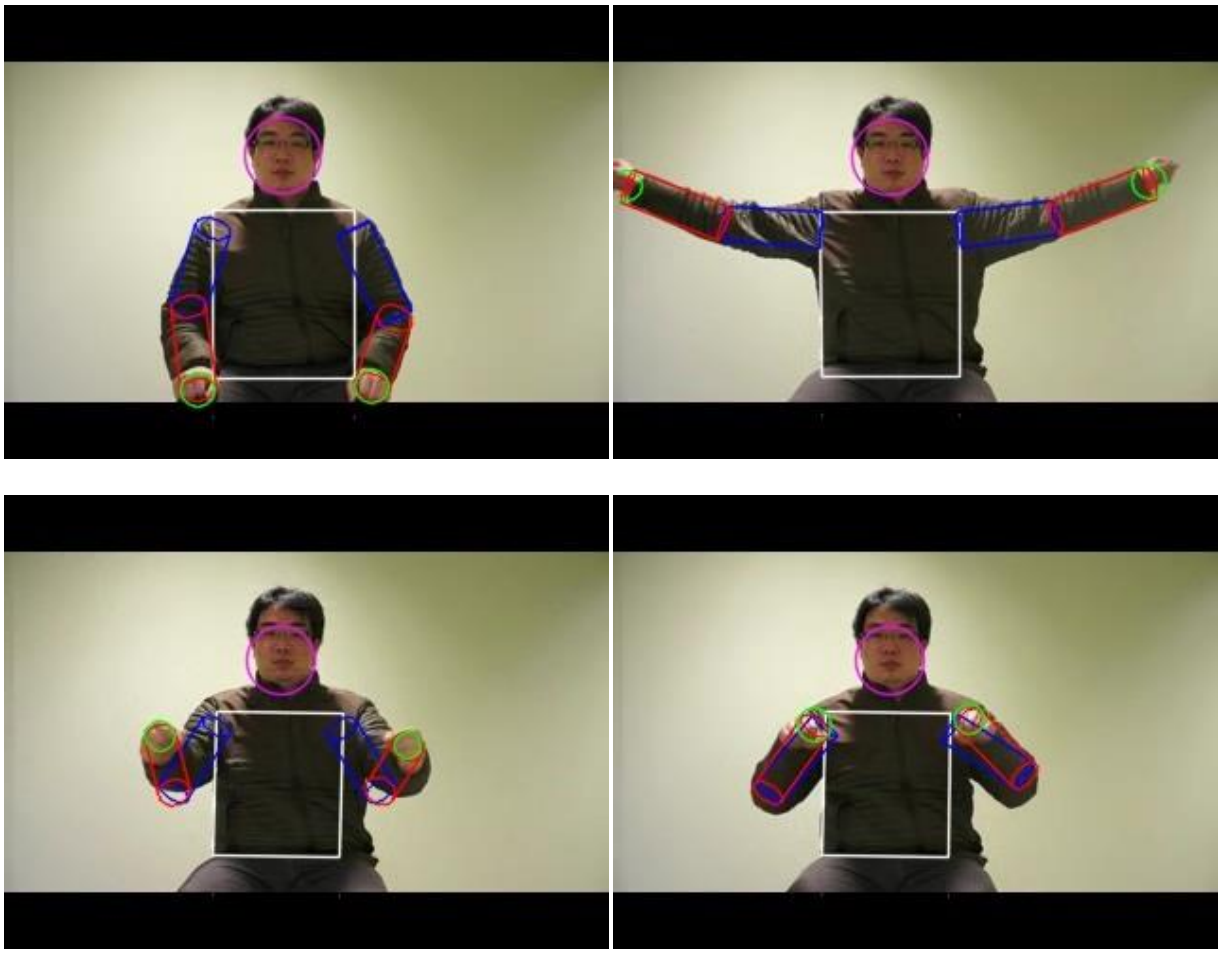
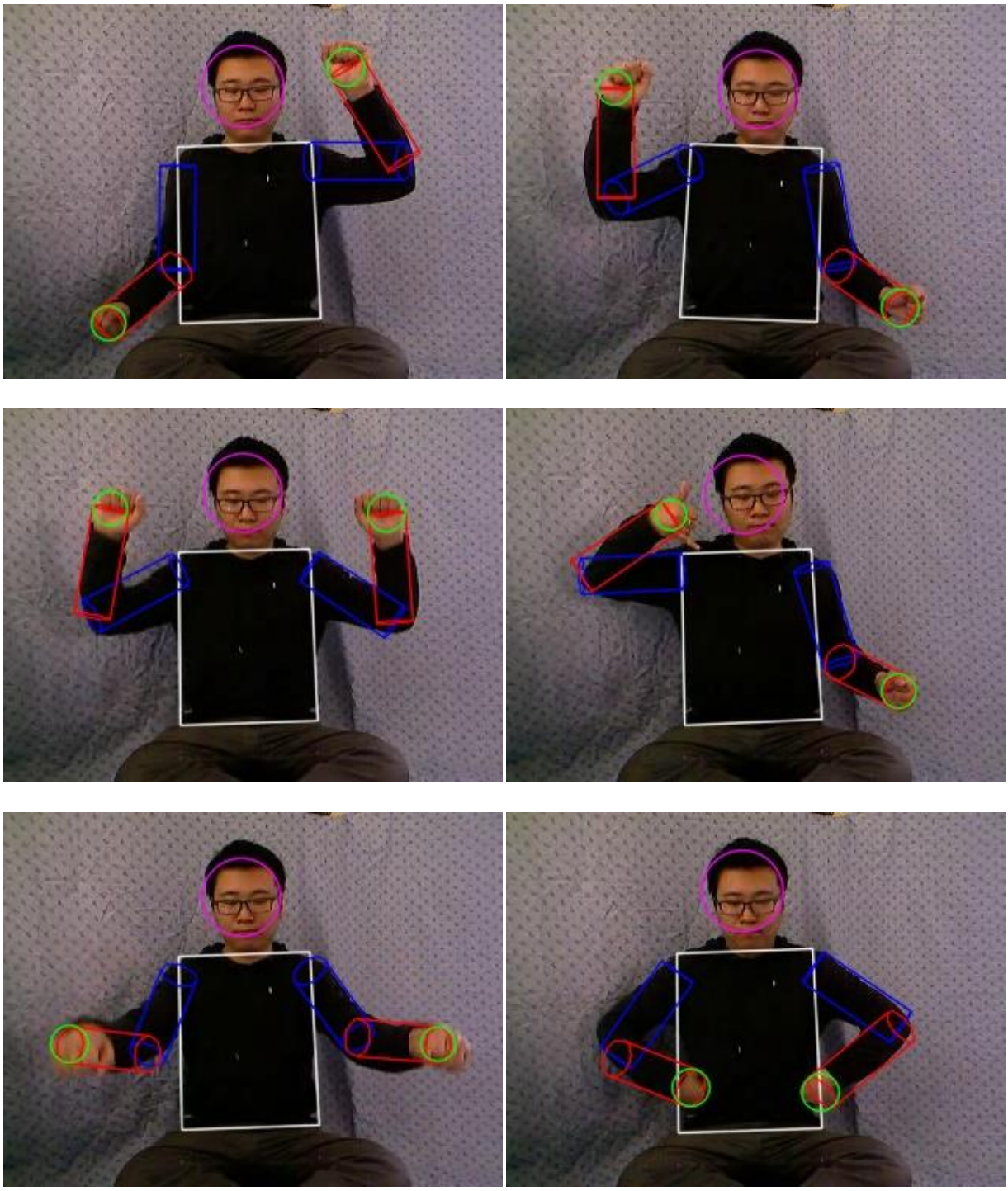


Figure 4.2 - 10: Dataset 7 Results

Figure 4.2 - 11 shows ground-truth dataset-1 results from our system.



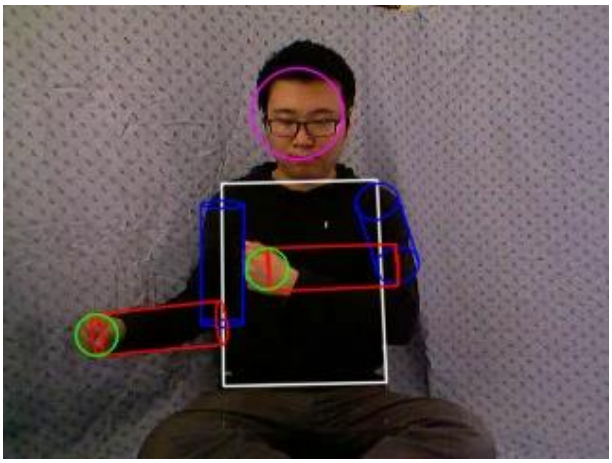
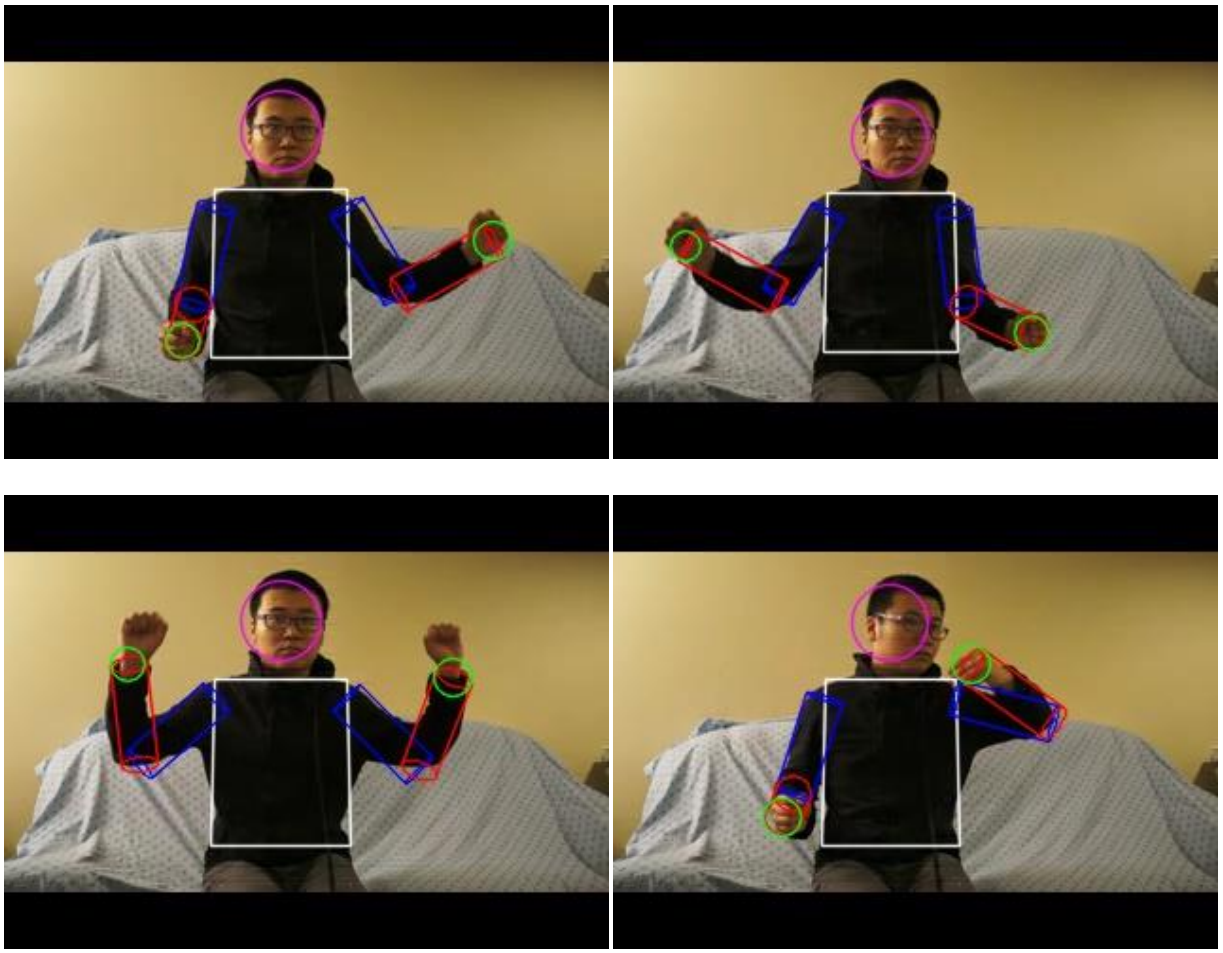


Figure 4.2 - 11: Ground-Truth Dataset 1 Results

Figure 4.2 - 12 shows ground-truth dataset-2 results from our system.



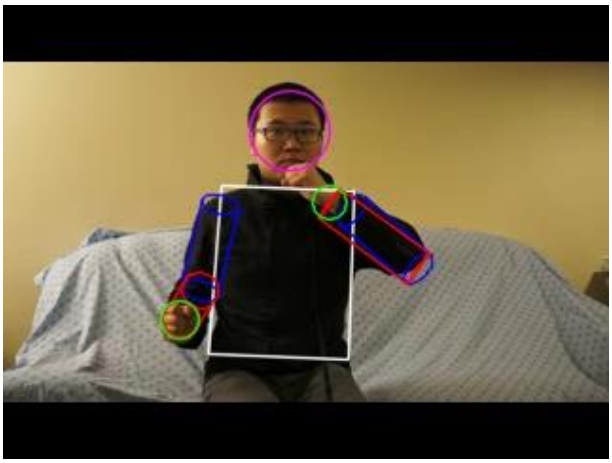
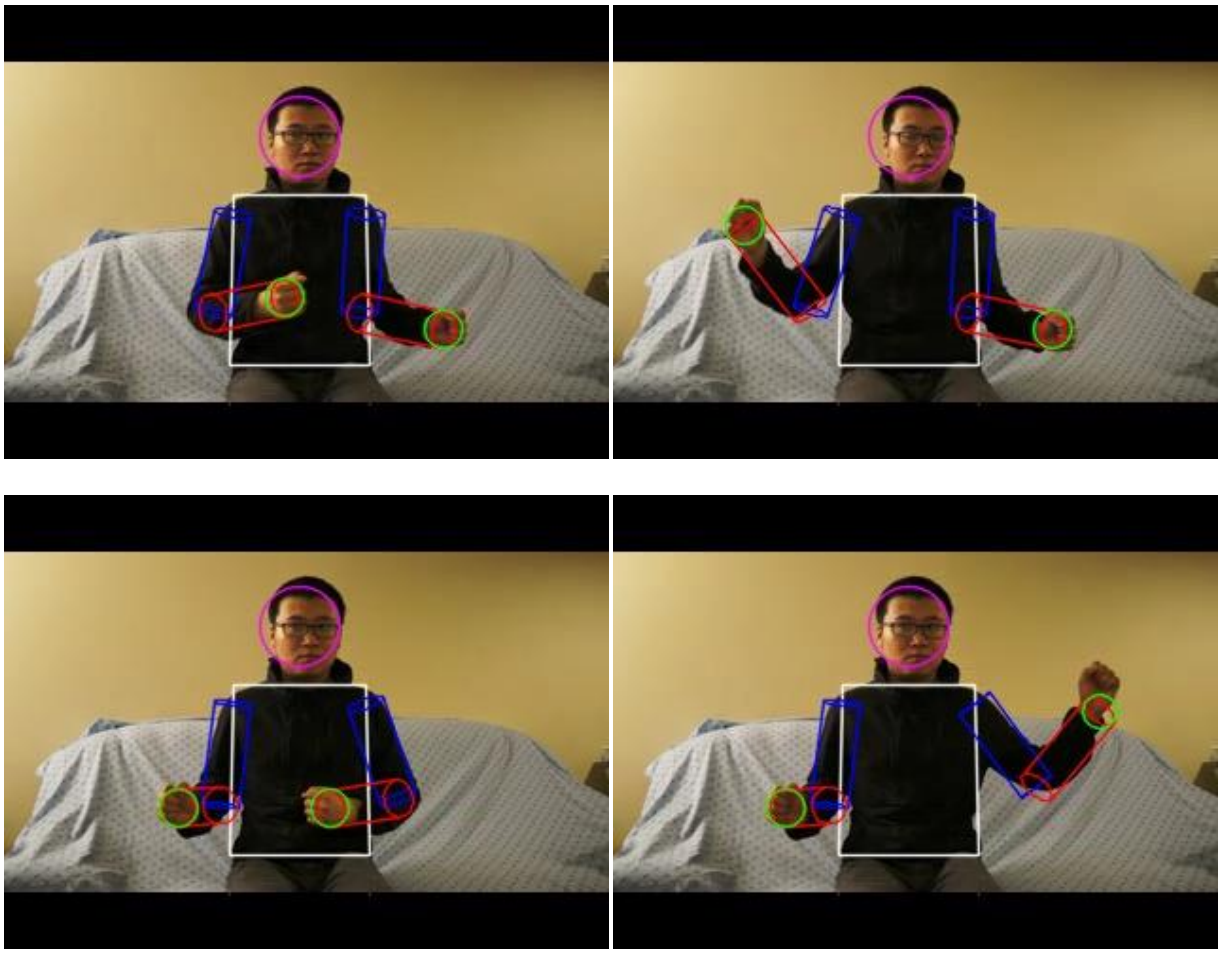


Figure 4.2 - 12: Ground-Truth Dataset 2 Results

Figure 4.2 - 13 shows ground-truth dataset-3 results from our system.



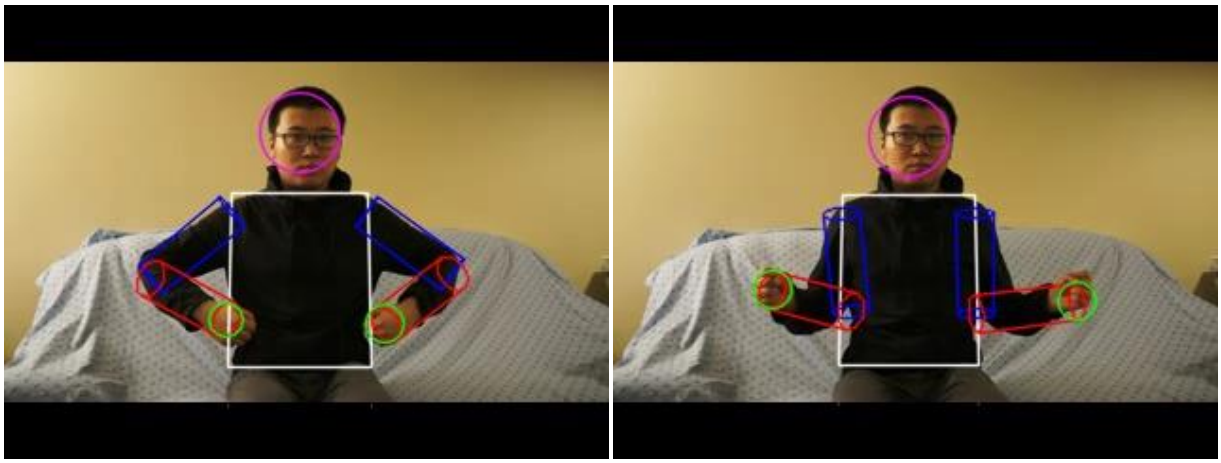
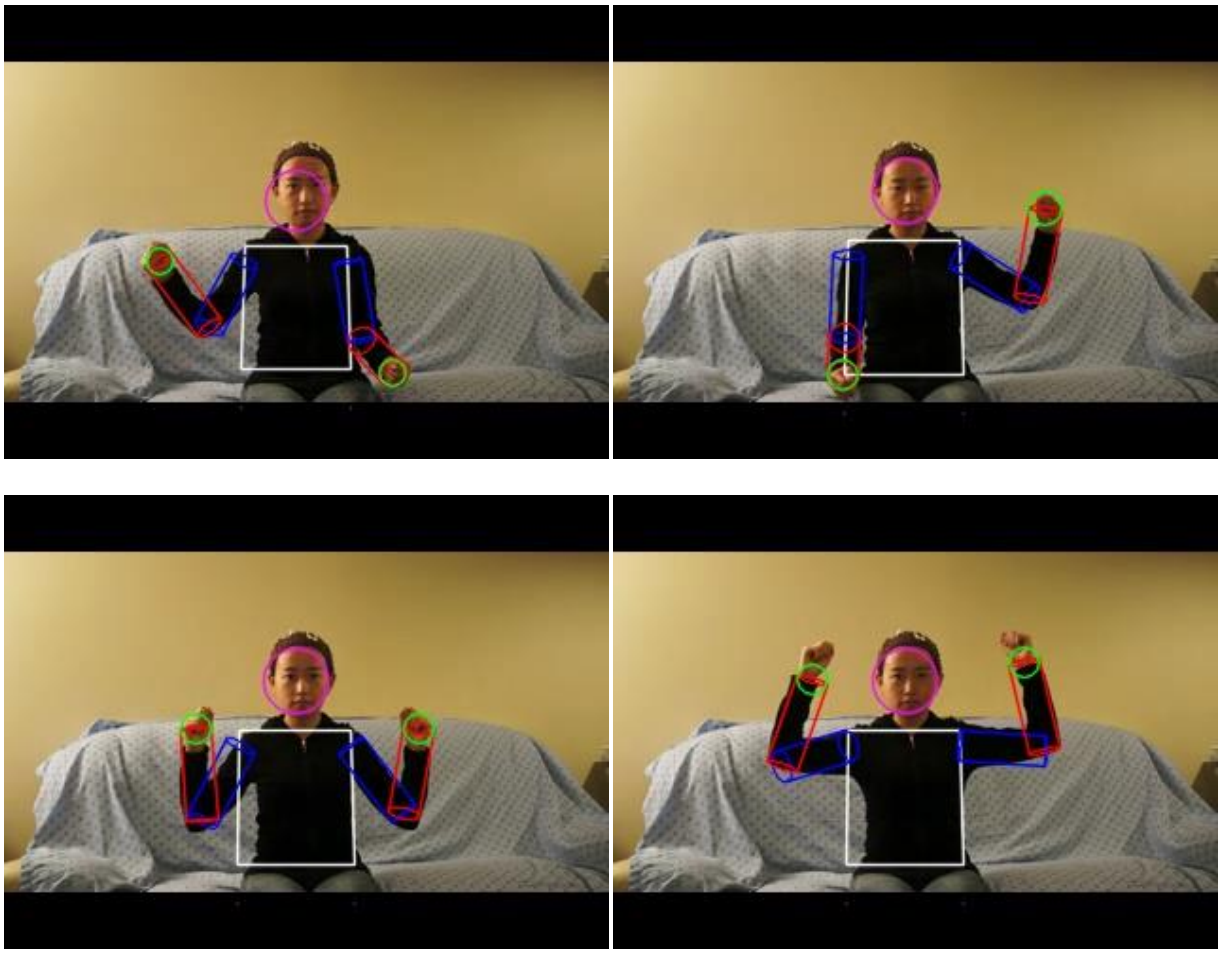


Figure 4.2 - 13: Ground-Truth Dataset 3 Results

Figure 4.2 - 14 shows ground-truth dataset- 4 results from our system.



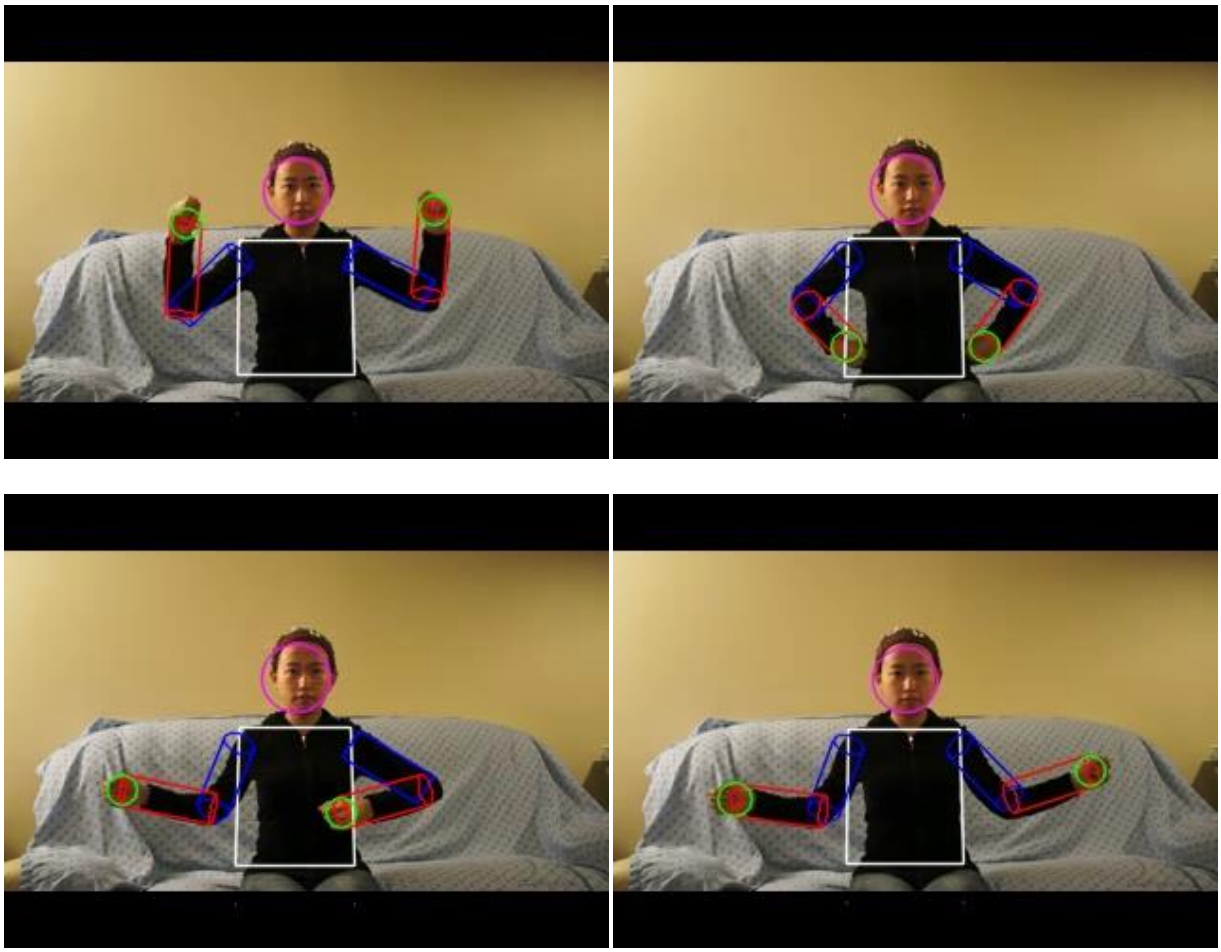
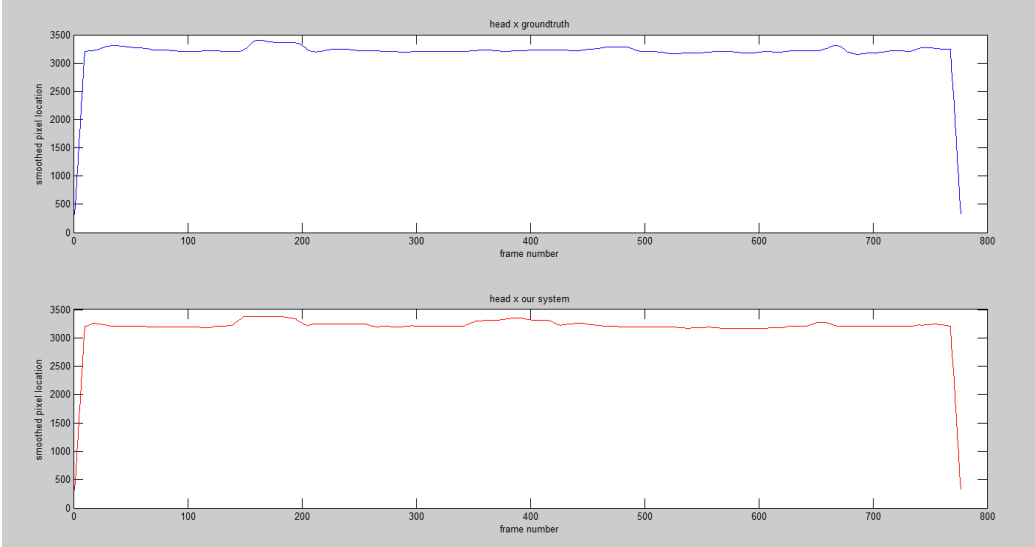
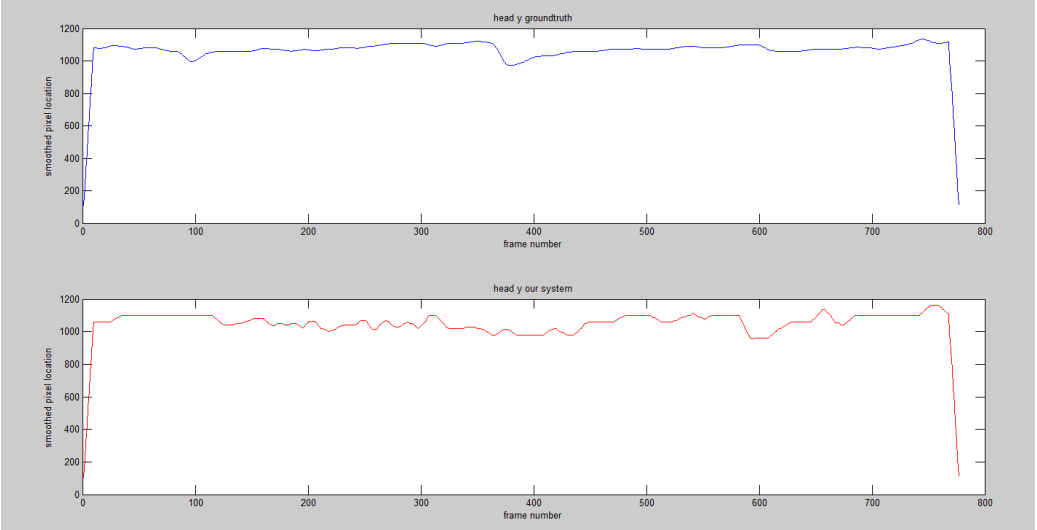
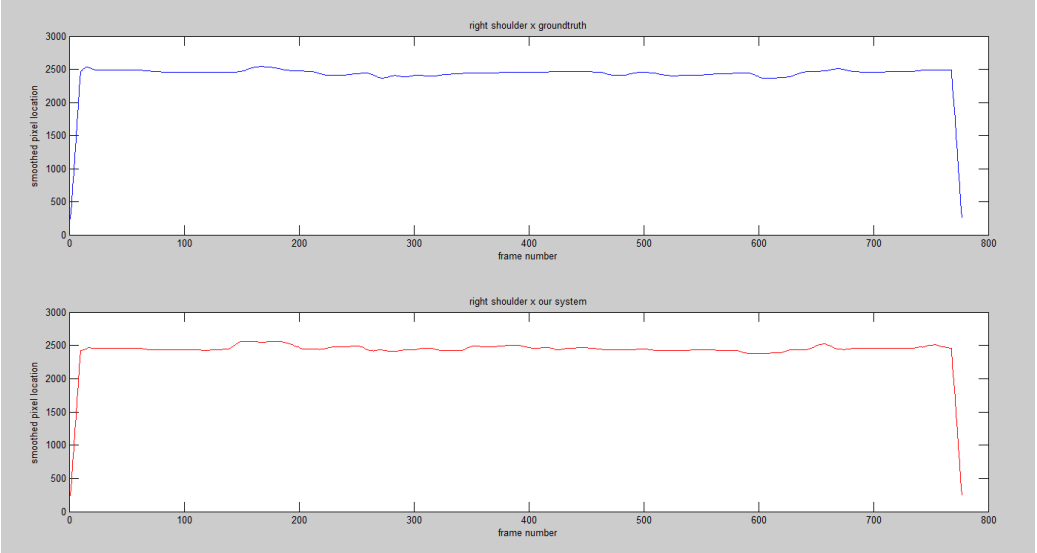


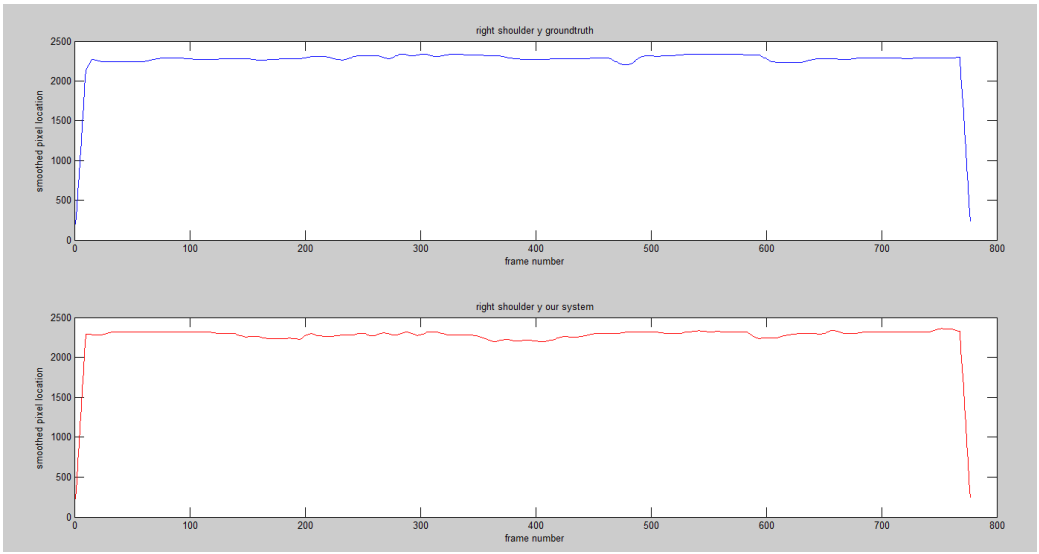
Figure 4.2 - 14: Ground-Truth Dataset 4 Results

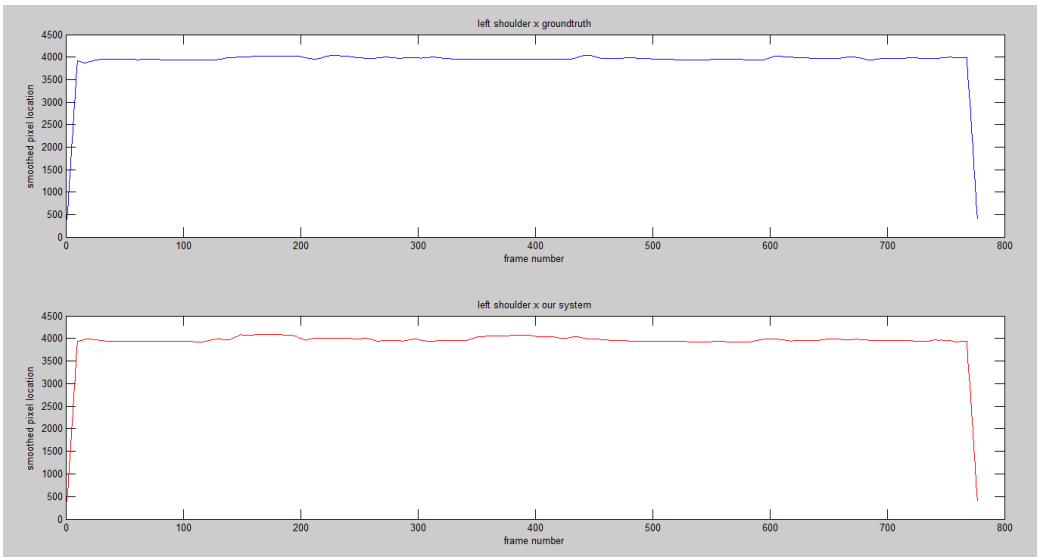
Figure 4.2 - 15 shows graphical pixel location comparison between our system and ground-truth for each body parts' x, y position from ground-truth dataset 1. In those picture results, the first graph with blue color means the ground-truth of a specific body part's x or y location; the second graph with red color means x, y location from our system. The order is head-x, head-y, right-shoulder-x, right-shoulder-y, left-shoulder-x, left-shoulder-y, right-elbow-x, right-elbow-y, left-elbow-x, left-elbow-y, right-hand-x, right-hand-y, left-hand-x, and left-hand-y.

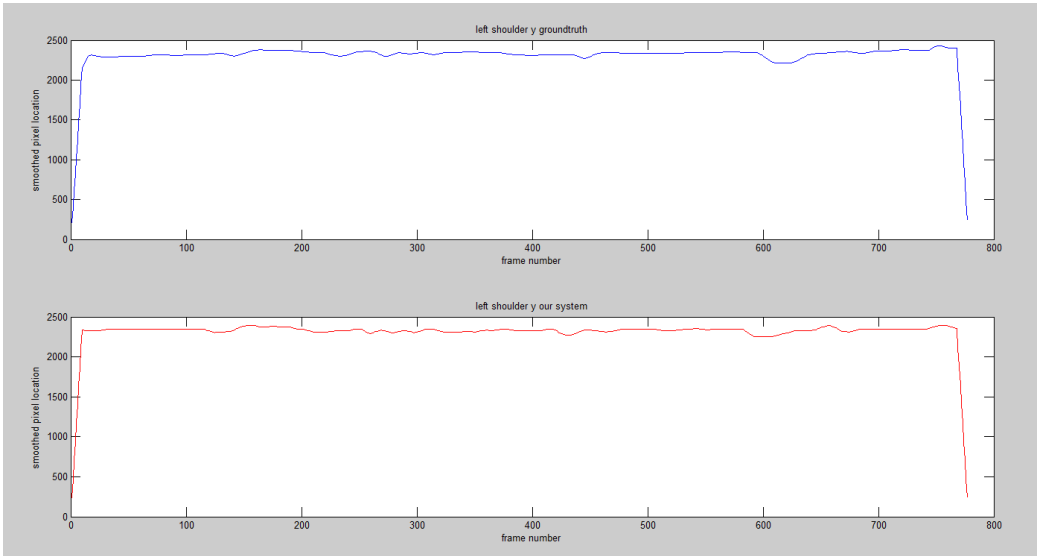

head x groundtruth
smoothed pixel location
frame number
head x our system
smoothed pixel location
frame number

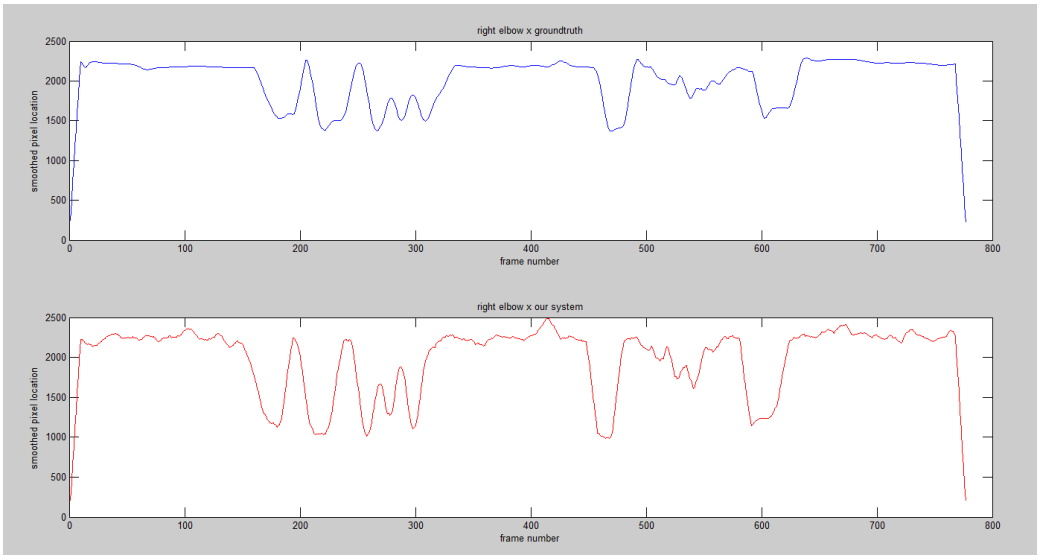

head y groundtruth
smoothed pixel location
frame number
head y our system
smoothed pixel location
frame number

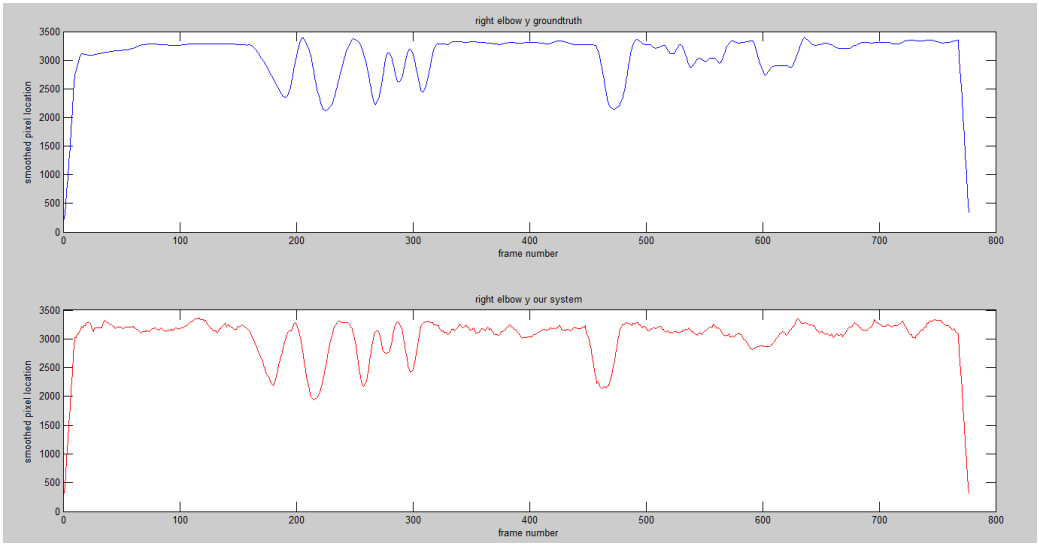

right shoulder x groundtruth
smoothed pixel location
frame number
right shoulder x our system
smoothed pixel location
frame number

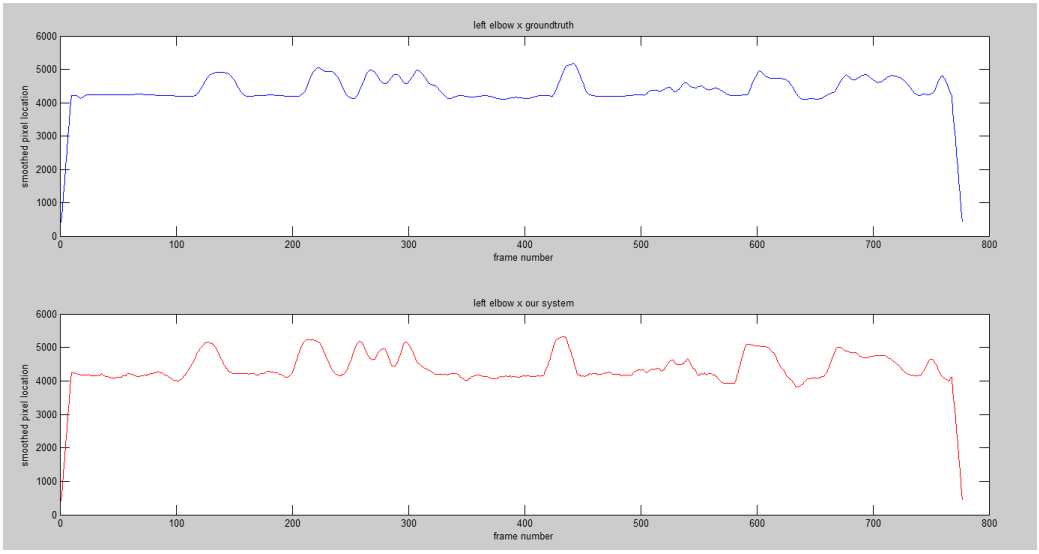

right shoulder y groundtruth
smoothed pixel location
frame number
right shoulder y our system
smoothed pixel location
frame number

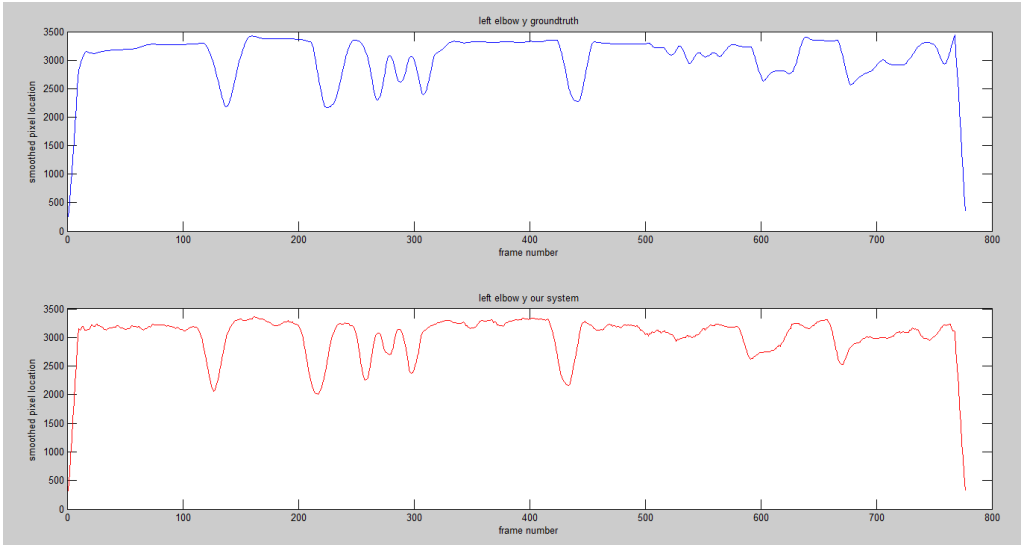

left shoulder x groundtruth
smoothed pixel location
frame number
left shoulder x our system
smoothed pixel location
frame number

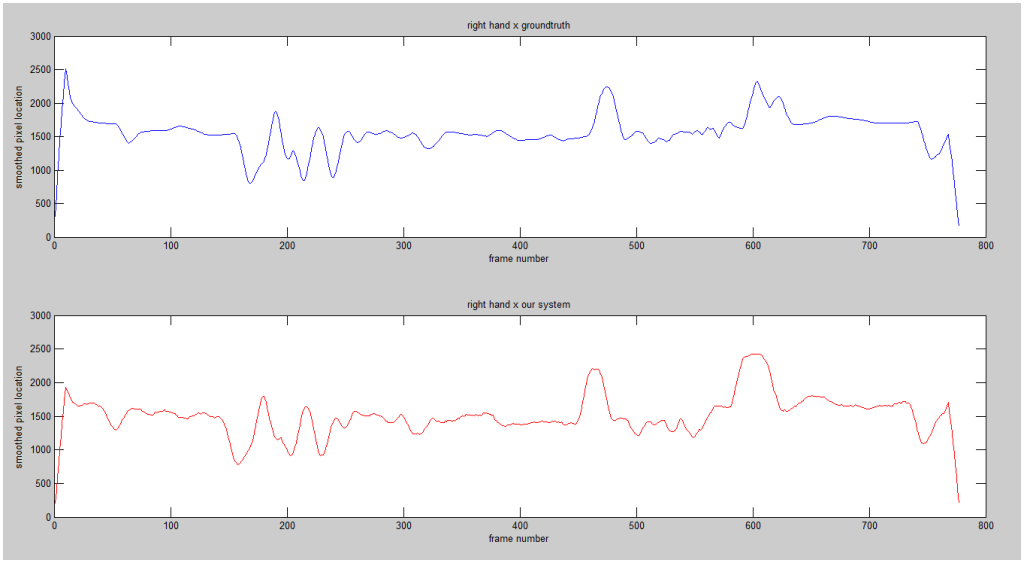

left shoulder y groundtruth
smoothed pixel location
frame number
left shoulder y our system
smoothed pixel location
frame number

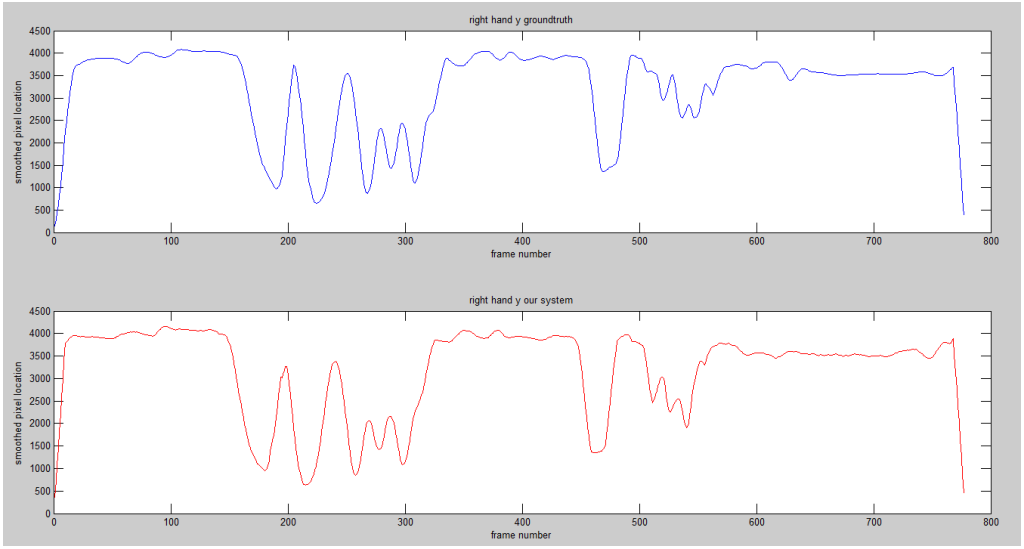

right elbow x groundtruth
smoothed pixel location
frame number
right elbow x our system
smoothed pixel location
frame number


right elbow y groundtruth
smoothed pixel location
frame number
right elbow y our system
smoothed pixel location
frame number


left elbow x groundtruth
smoothed pixel location
frame number
left elbow x our system
smoothed pixel location
frame number


left elbow y groundtruth
smoothed pixel location
frame number
left elbow y our system
smoothed pixel location
frame number


right hand x groundtruth
smoothed pixel location
frame number
right hand x our system
smoothed pixel location
frame number


right hand y groundtruth
smoothed pixel location
frame number
right hand y our system
smoothed pixel location
frame number

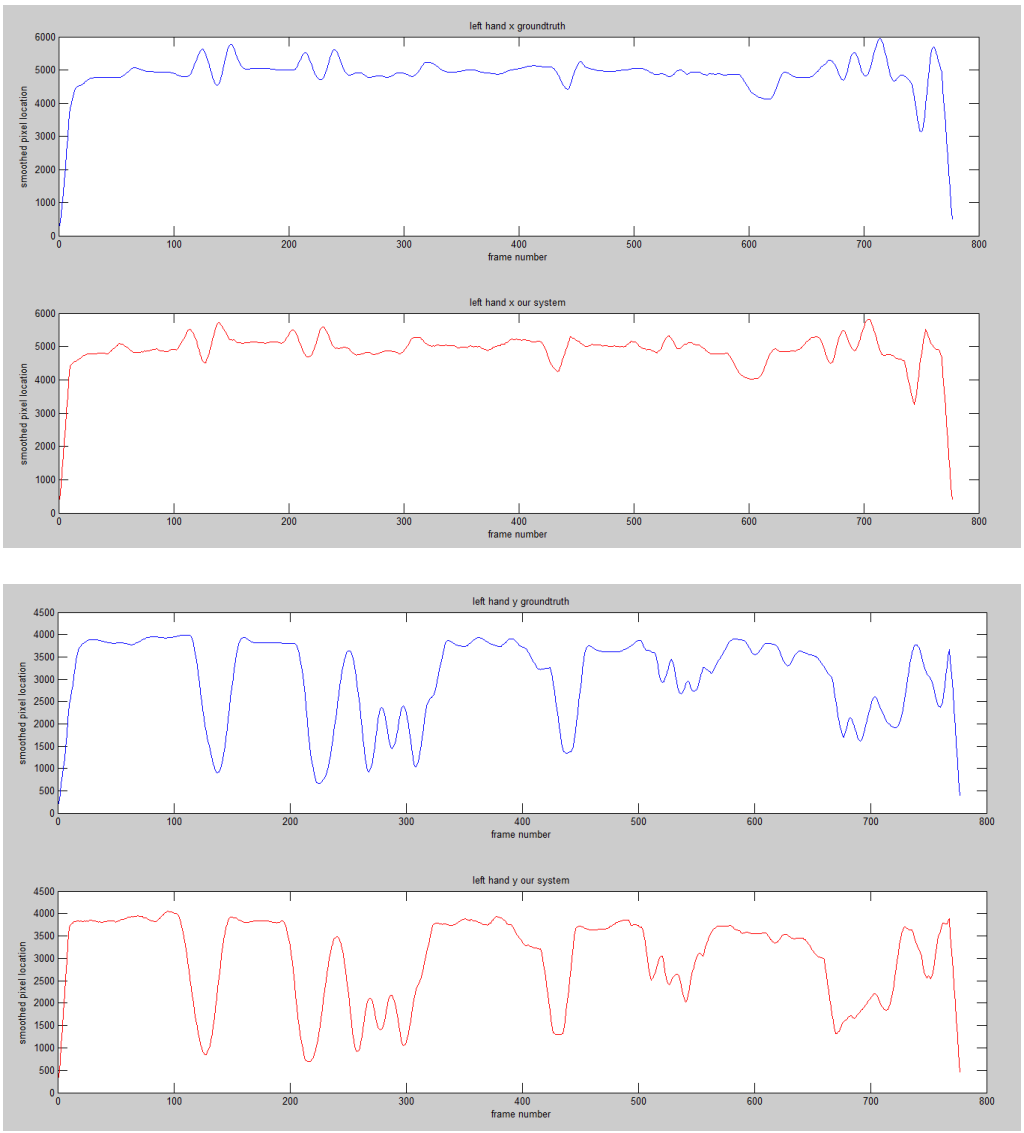
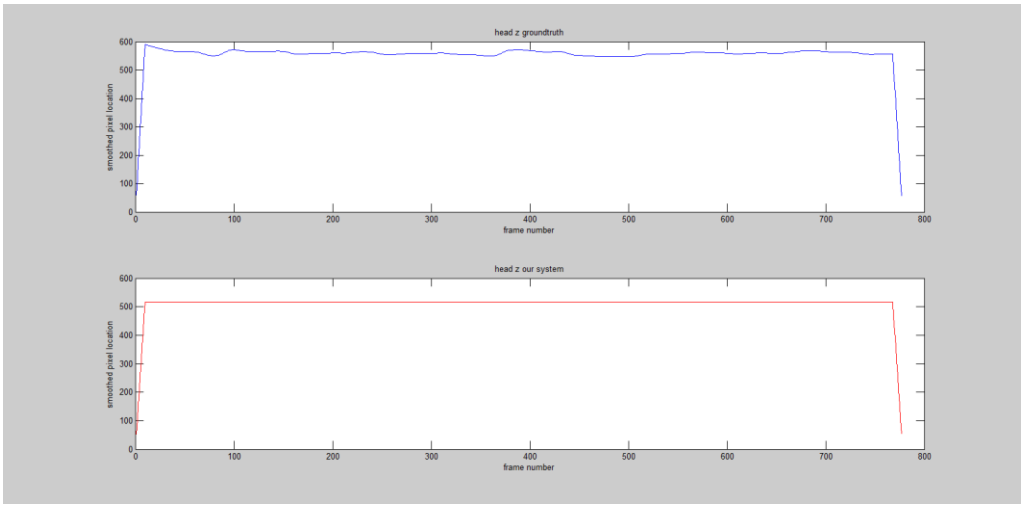
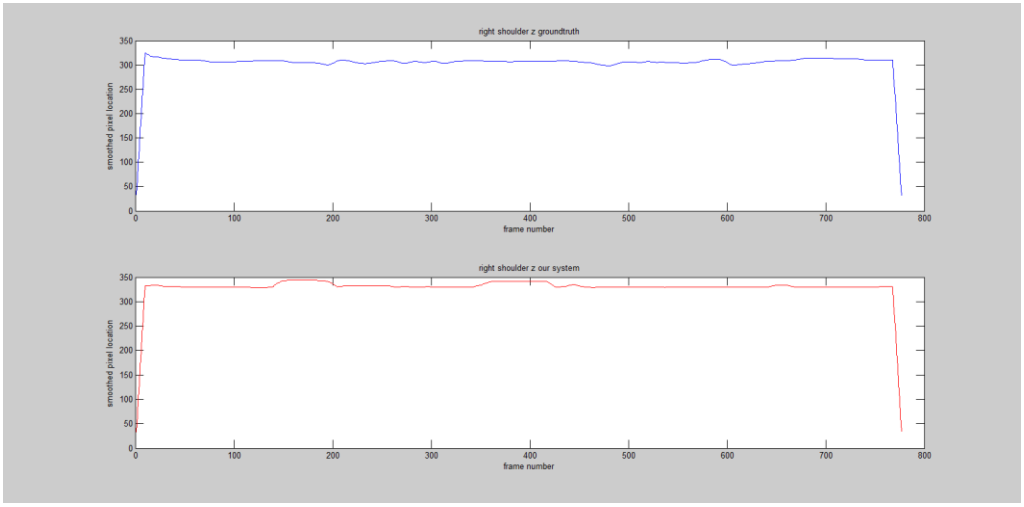
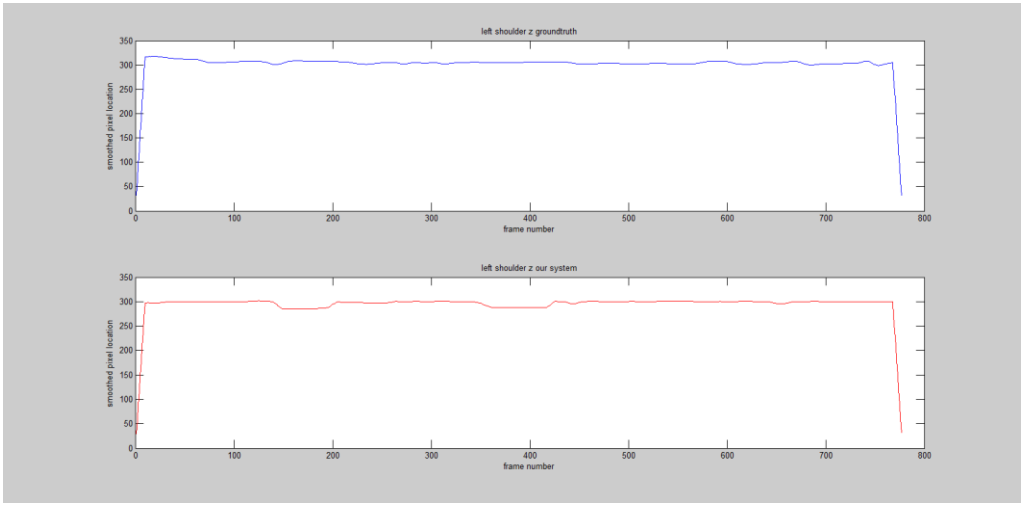


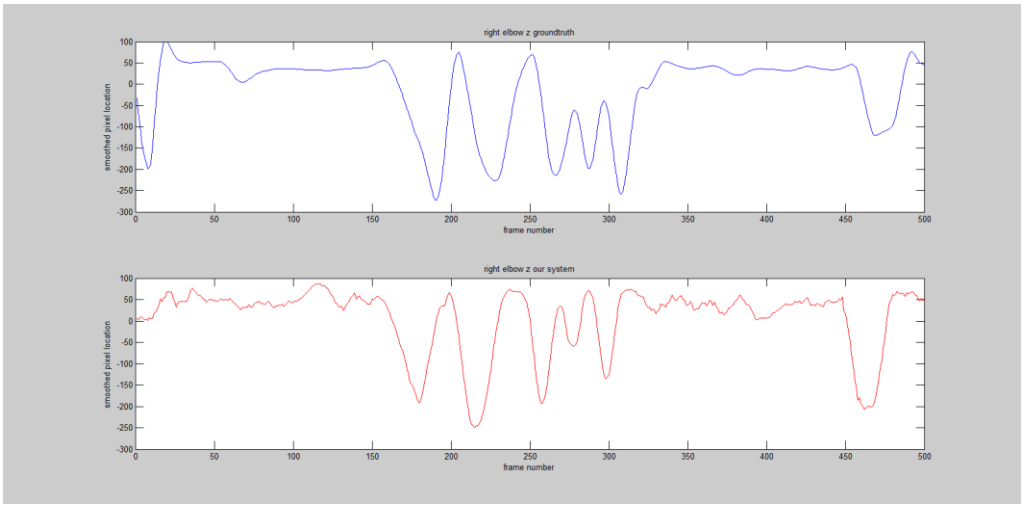
Figure 4.2 - 15: Pixel Location Comparison

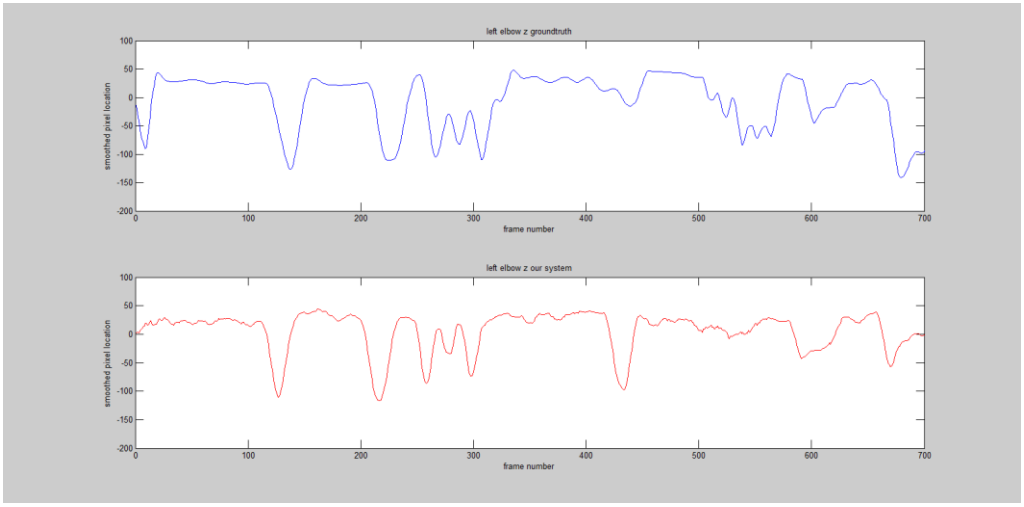
Figure 4.2 - 16 shows the z location comparison for each body part from ground-truth dataset 1.

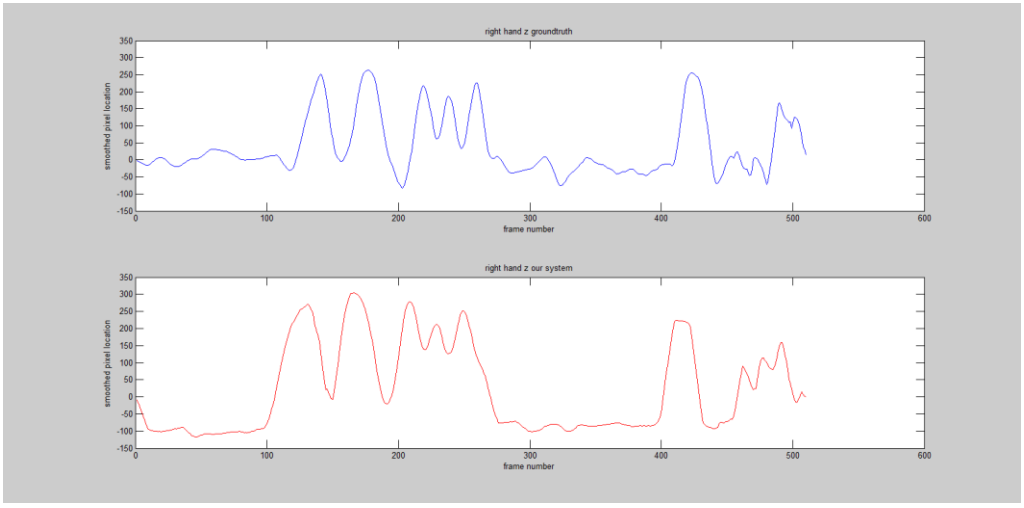

head z groundtruth
smoothed pixel location
frame number
head z our system
smoothed pixel location
frame number


right shoulder z groundtruth
smoothed pixel location
frame number
right shoulder z our system
smoothed pixel location
frame number


left shoulder z groundtruth
smoothed pixel location
frame number
left shoulder z our system
smoothed pixel location
frame number


right elbow z groundtruth
smoothed pixel location
frame number
right elbow z our system
smoothed pixel location
frame number


left elbow z groundtruth
smoothed pixel location
frame number
left elbow z our system
smoothed pixel location
frame number


right hand z groundtruth
smoothed pixel location
frame number
right hand z our system
smoothed pixel location
frame number

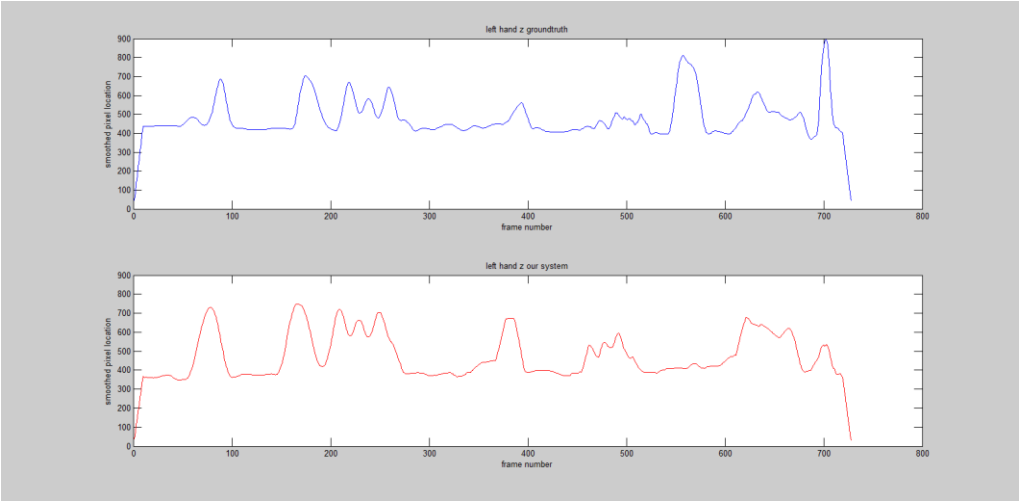
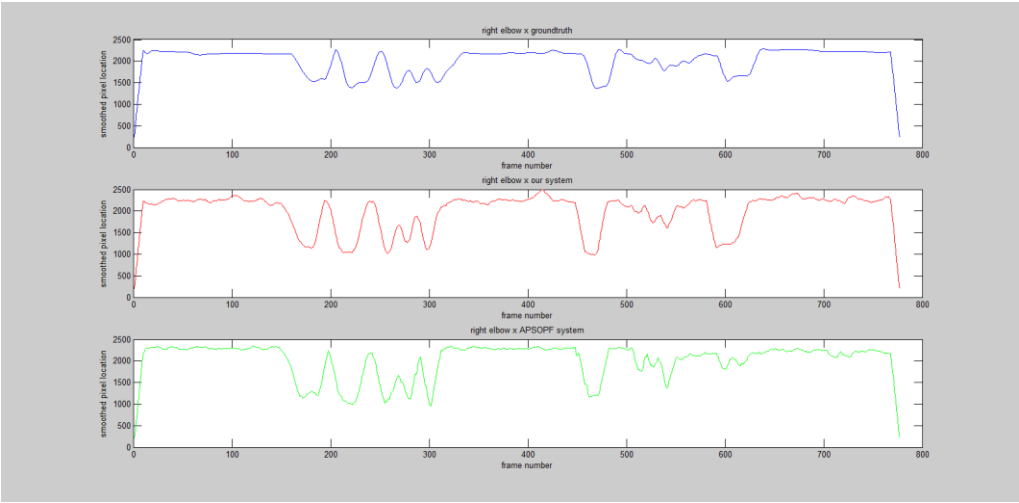
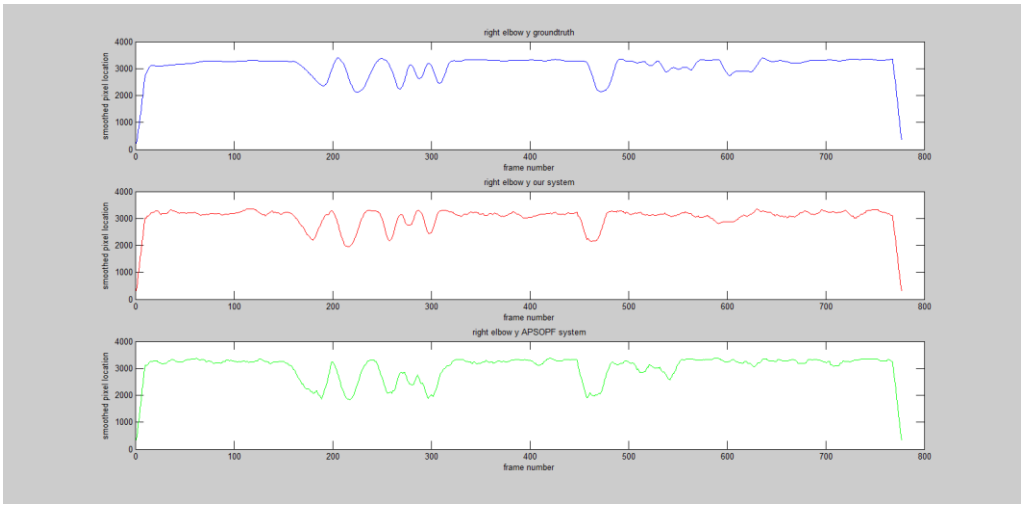
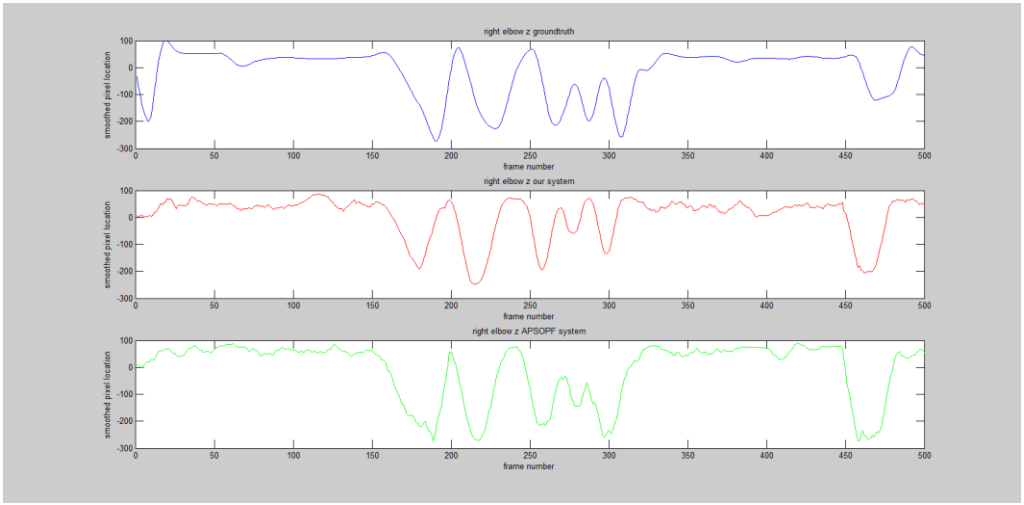


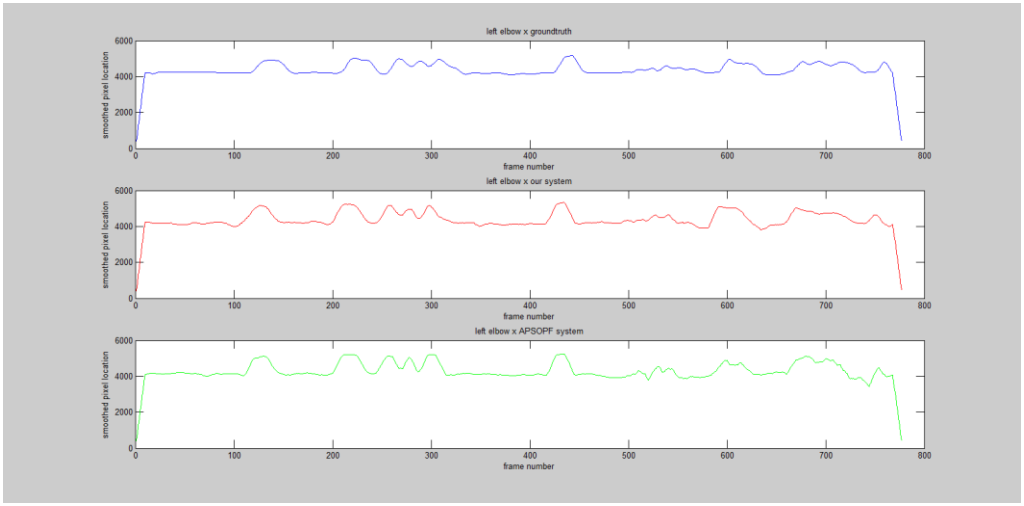
Figure 4.2 - 16: Pixel z-Location Comparison

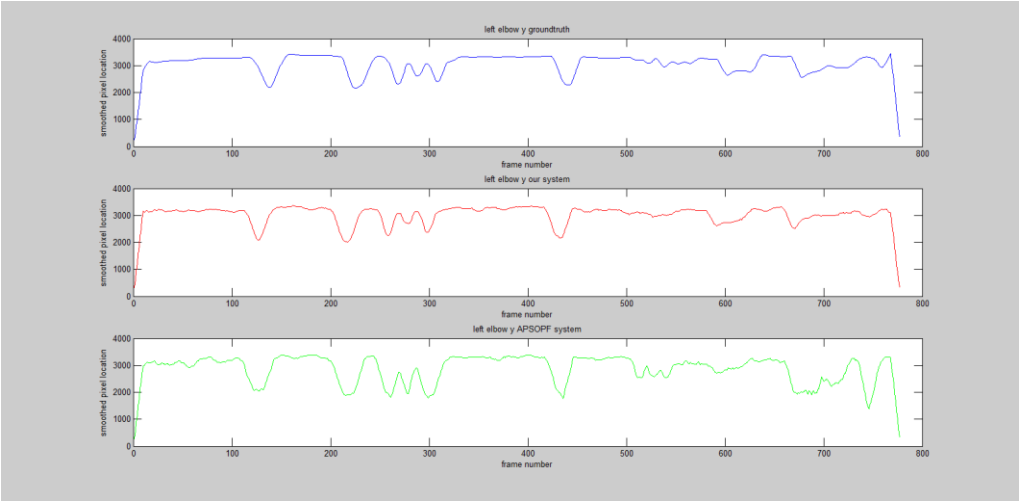
Figure 4.2 - 17 shows example picture results for the x, y, z location comparison for right elbow and left elbow from ground-truth dataset 1. The detail for all body parts comparison is shown in Table 4.2 - 1 from all ground-truth datasets. Table 4.2 - 1 shows the **average absolute difference** of pixel position between our system results and ground-truth results and between APSOPF system results [2] and ground-truth results for the total four ground-truth datasets. Bold color in this table represents the better value.




right elbow y groundtruth
smoothed pixel location
frame number
right elbow y our system
smoothed pixel location
frame number
right elbow y APSOPF system
smoothed pixel location
frame number


right elbow z groundtruth
smoothed pixel location
frame number
right elbow z our system
smoothed pixel location
frame number
right elbow z APSOPF system
smoothed pixel location
frame number


left elbow x groundtruth
smoothed pixel location
frame number
left elbow x our system
smoothed pixel location
frame number
left elbow x APSOPF system
smoothed pixel location
frame number



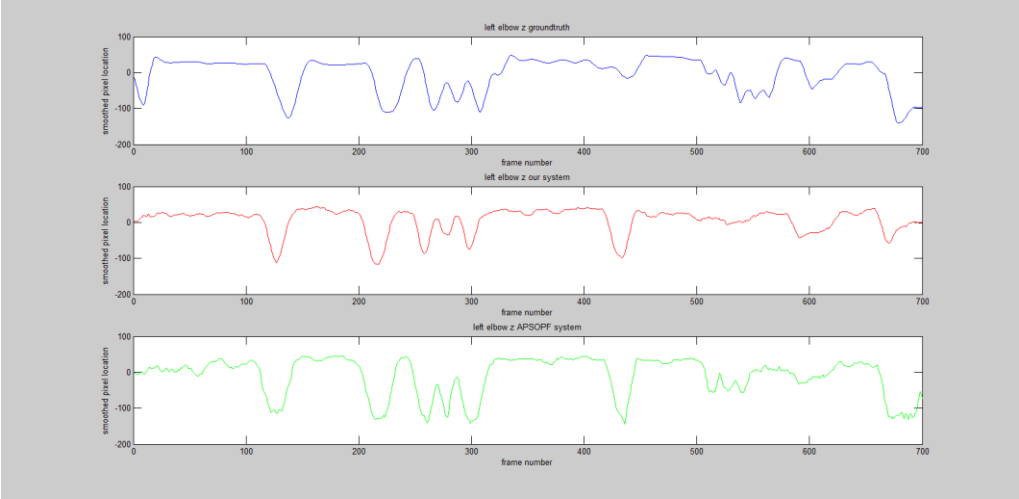


Figure 4.2 - 17: Comparison with APSOPF System

Body Parts Name	Ground-truth Dataset 1		Ground-truth Dataset 2		Ground-truth Dataset 3		Ground-truth Dataset 4	
	Mine	APSOPF	Mine	APSOPF	Mine	APSOPF	Mine	APSOPF
Head x	**3.6068**	4.0755	**1.0451**	12.4312	**1.0206**	13.9396	**1.5832**	7.8534
Head y	**3.8307**	4.1302	**4.0203**	12.2596	**1.1636**	1.2813	**1.5766**	9.3180
Head z	**4.3954**	4.3973	1.6863	**1.5428**	1.0311	**1.0227**	0.6917	**0.6826**
Right shoulder x	**2.7799**	3.5820	**2.6429**	11.5207	**2.7531**	12.5329	**1.4314**	7.7140
Right shoulder y	**4.0573**	4.5677	**2.9120**	3.0068	**2.5275**	5.2355	**2.2058**	6.7578
Right shoulder z	**2.5079**	3.9415	**0.1363**	0.2172	**0.0548**	0.3533	0.6825	**0.4344**

Left shoulder x	**3.6901**	3.9297	**1.7512**	4.8802	**1.7248**	9.7248	**1.4164**	14.2375
Left shoulder y	**3.0768**	3.1576	**2.7314**	9.4221	**1.9610**	3.9503	**3.2801**	8.2076
Left shoulder z	**0.74**	0.9076	**0.6657**	0.7381	**0.1019**	0.1274	**0.0916**	0.3125
Right elbow x	**14.8256**	19.6077	**9.7024**	17.5791	**16.2936**	18.9694	**11.6259**	14.8711
Right elbow y	**15.1691**	19.8877	**9.4879**	19.7802	**9.1728**	15.4832	**8.5339**	15.5583
Right elbow z	**4.9871**	5.8261	**10.6718**	11.3125	**9.9295**	10.1387	**12.3211**	13.6335
Left elbow x	**13.5152**	22.2602	**9.8624**	19.3175	**10.6047**	19.8846	**11.2103**	13.7233
Left elbow y	**11.3170**	26.6248	**11.5764**	20.7694	**10.5061**	16.8532	**11.0510**	12.7950
Left elbow z	**2.6681**	2.7817	**10.6227**	11.5938	**9.7189**	10.5908	**12.2973**	12.8490
Right hand x	**17.2285**	17.9075	14.3782	**13.9640**	**18.3128**	18.7982	**13.5363**	13.8010
Right hand y	**14.7160**	15.5694	**17.5353**	18.4487	**15.9553**	18.3544	16.5845	**16.1127**
Right hand z	**7.2437**	7.3046	**10.0107**	10.2195	**7.7432**	7.8136	**10.9637**	11.4360
Left hand x	**13.19**	15.4016	**17.0563**	20.5576	**12.8213**	13.5347	**12.3011**	12.7149
Left hand y	**18.9657**	19.7609	**18.0508**	18.0508	**12.4545**	12.6857	**15.3785**	16.4146
Left hand z	**7.6084**	7.6757	**8.8509**	9.1804	**7.4333**	7.6138	**9.8186**	10.055
Total average x	**8.639625**				13.54693			
Total average y	**8.92135**				12.6587			
Total average z	**5.5598**				5.882225			
Frame rate	**4.9979**	0.4089	**5.3529**	0.4438	**3.4187**	0.4475	**5.3389**	0.4298

Table 4.2 - 1: Average Absolute Difference of Pixel Position

Table 4.2 - 2, Table 4.2 - 3, and Table 4.2 - 4 show the different sigma values comparison examples for right elbow, left elbow, and right hand; and explanations will be given in the next section.

Right elbow	x	y
My error	14.8256	15.1691
Dr. Wang's original error	**19.6077**	**19.8877**
Sigma 0.1	24.4947	22.4512
Sigma 0.15	21.6794	23.8047
Sigma 0.2	19.4472	21.4749
Sigma 0.25	22.5923	22.7916
Sigma 0.3	19.5897	20.9894
Sigma 0.5	20.8852	24.2269
Sigma 0.8	20.2493	21.8839
Sigma 0.9	20.4763	19.8417
Sigma 0.95	21.3153	20.6755
Sigma 1.0	19.0910	21.7493
Sigma 1.02	21.0488	20.8443
Sigma 1.05	20.4024	20.9815
Sigma 1.1	21.2573	21.9261
Sigma 1.2	21.2810	21.9182
Sigma 1.5	21.5686	21.4485

Table 4.2 - 2: Right Elbow Sigma Comparison

Left elbow	x	y
My error	13.5152	11.3170
Dr. Wang's original error	**22.2602**	**26.6248**
Sigma 0.1	18.624	36.876
Sigma 0.15	22.0488	51.8153
Sigma 0.2	30.9090	41.0712
Sigma 0.25	22.3364	47.3615
Sigma 0.3	31.6372	38.4538
Sigma 0.5	24.7322	28.5462
Sigma 0.8	**19.1649**	**22.3536**
Sigma 0.9	25.4631	26.8945
Sigma 0.95	24.3997	24.7282
Sigma 1.0	22.6082	25.3536
Sigma 1.02	24.4683	26.2375
Sigma 1.05	26.0462	25.7942
Sigma 1.1	24.7902	27.6016
Sigma 1.2	25.6240	28.0607
Sigma 1.5	25.8166	26.5805

Table 4.2 - 3: Left Elbow Sigma Comparison

Right hand	x	y
My error	17.2285	14.7160
Dr. Wang's original error	**17.9075**	**15.5694**
Sigma 0.1	17.9142	15.7005
Sigma 0.15	18.9380	21.8404
Sigma 0.2	18.7902	20.2098
Sigma 0.25	19.2124	21.5053
Sigma 0.3	18.5475	19.7322
Sigma 0.5	19.6794	14.9908
Sigma 0.8	19.1834	14.9327
Sigma 0.9	17.6451	19.6425
Sigma 0.95	19.5237	20.6293
Sigma 1.0	18.0778	14.8720
Sigma 1.02	19.2018	20.5211
Sigma 1.05	17.7507	19.6108
Sigma 1.1	**17.8536**	**13.6478**
Sigma 1.2	18.996	14.785
Sigma 1.5	19.1596	15.2335

Table 4.2 - 4: Right Hand Sigma Comparison

4.3 Explanations of Results

From Figure 4.2 - 15 and Table 4.2 - 1 show that there is not much difference between our system and ground-truth for head, right shoulder, and left shoulder; since it is front-view, there is not much movement for these three body parts. For the rest of body parts, there are minor difference between our system and ground-truth. From Figure 4.3 – 1, the right elbow can be seen that there is a minor difference. The reason can be seen from Figure 4.3 – 2. These are not the same frame and they are chosen in order to provide a clear explanation. From the front-view, right arm are very close to the torso, so from a silhouette view Figure 4.3 – 2, the torso silhouette affects the estimation about the right arm (self-occlusion). Possible solution improvement is mentioned in section 5.2 future work.

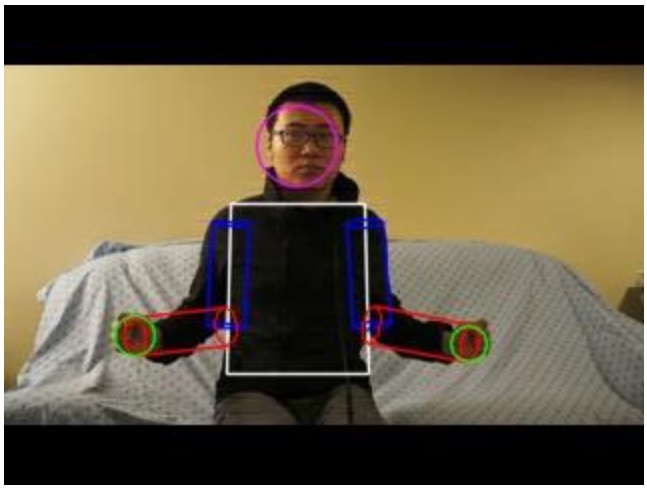


Figure 4.3 – 1: Minor Difference Example

Figure 4.3 – 2: Minor Difference Silhouette View Example

Figure 4.3 - 3 shows images extracted from paper [1] and [2] in order to show their methods do have limitations, and to prove our implementation for their methods is honest and reasonable.

Figure 4.3 – 3: Extracted Images

Paper [1] is implemented for comparison. The results can be seen from Figure 4.2 – 2. It is obvious that our results are better than [1]. In this paper, the author Alex Ke assumes that all human movement operations are straight to simplify the optimization problem in order to achieve real-time analysis. For example, arms have to move straight, and they cannot bend, see Figure 4.2 – 2. Another reason is that [1] uses downhill simplex optimization algorithm to search for the pose. It is a simple operation algorithm and cannot handle complex movement, such as arm bending. In addition, it is very obvious our results are better so that pixel difference comparison is not shown in Table 4.2 – 1.

Paper [2] is implemented for comparison. The results can be seen from Figure 4.2 – 3. [2]'s APSOPF algorithm can be divided into two parts, APSO and PF (particle filter). [2] uses particle filter to train datasets in order to obtain accurate angles, and then uses these angles to calculate an angle range for constraints. [2] uses this particle filter trained constraint as sigma values for the Gaussian re-distribution particles. However, our system uses constrained uniform distribution instead of Gaussian, because if the Gaussian distribution is constrained, there will be some distributed particles accumulating on the boundary, which are wasted. This is why we choose to use uniform distribution. Instead of training an angle range, we simplify the work and uses reasonable angle constraints. Table 4.2 – 2, Table 4.2 – 3, and Table 4.2 – 4 show examples to prove that sigma values (represent the angle range) do not provide an effective improved results as long as they are in a reasonable range. Our system contains six image features, while [2] only uses two image features silhouette and edge. Therefore, our results are better than [2], see Table 4.2 – 1, even when our search algorithm APSO is simpler than [2]'s APSOPF. Our frame rate is also higher than [2]'s frame rate. Mainly, this is because the motion feature that is used, see details in section 2.5 and section 3.4.3.

4.4 Chapter Conclusion

This chapter describes the specific source of experimental data and the reason of choosing this source. Also, this chapter introduces the typicality of the selected video data. Through the experimental verification of this chapter, it proves the 3D human motion tracking framework based on annealing particle swarm optimization can achieve monocular videos 3D human body pose recovery. Through the comparative analysis of the experimental results, the proposed results are closer to the ground-truth than results of [1] and [2]. This chapter also makes an explanation why proposed system has better performance. By comparing paper [2], it proves the importance and usefulness of added image features; meanwhile, it explains the strong constraint is conducive to the accuracy of the results. Moreover, this chapter proves that when there is self-occlusion or occlusion produced by in the human motion process, added image features and precise constraints can provide very accurate results.

Chapter 5:
CONCLUSION AND OUTLOOK

5.1 Paper Summary

The 3D human motion tracking is a hot computer vision field and a very challenging research direction. The estimated human motion pose data can be further used for emotional recognition, action recognition, identity recognition and other research work.

Annealing particle swarm optimization is introduced to the 3D human tracking in this paper. It also talks about human motion constraints. After the theoretical in-depth study and experimental verification based on real data, annealing particle swarm optimization is decided to be suitable for solving high-dimensional space search problem, and greatly improves the efficiency of the tracking. In addition, the added human motion constraints increase the tracking robustness. Comprehensive select silhouette, edge, motion, skin, arm silhouette, and ratio six image features to construct cost function for judging the importance of each particle; and analyze the different purposes that these six features serve. Moreover, verify the proposed experimental results are closer to the ground-truth, which mean that they are better than other experimental results [1][2]. The designed system is also effective to self- occlusion situation.

5.2 Future Work

Single camera human pose estimation system cannot handle human's self-occlusion perfectly. The human motion constraint cannot solve the occlusion problem fundamentally, so how to combine physics, biology and other fields of knowledge to create an ideal human body model is a very meaningful work.

The mature technology of motion capture equipment can provide sufficient data for the human body pose initialization and follow-up experiments; however in reality, when a variety of human motion parameter recovery algorithms are used for ordinary monocular video, data initialization is very difficult to obtain; therefore, how to get the initial body pose data can be a in-depth research topic.

The designed system in this paper is very sensitive to illumination and some certain color i.e. red. Consider change the RGB color space to other color space, such as HSV, to prevent this sensitivity and improve the accuracy of pose estimation.

Due to the complexity of the structure of the human body, 3D human body pose data must be represented by high-dimensional model parameters. It is necessary to study more effective optimization and dimensionality reduction algorithms to improve the 3D human tracking algorithms and theory, in order to improve the search speed to achieve real-time requirements.

References

[1] Shian-Ru Ke, LiangJia Zhu, Jenq-Neng Hwang, Hung-I Pai, Kung-Ming Lan, Chih-Pin Liao, "Real-Time 3D Human Pose Estimation from Monocular View with Applications to Event Detection and Video Gaming," IEEE international conference on Advanced Video and Signal-Based Surveilliance (AVSS), Boston, U.S.A., August 29 - September 1, 2010.

[2] Xiangyang Wang, Xiang Zhou, Wanggen Wan, Xiaoqing Yu. "Articulated 3D Human Pose Estimation with Particle Filter based Particle Swarm Optimization", Audio Language and Image Processing (ICALIP), 2010 International Conference on, pp. 1094-1099, Nov.2010.

[3] Poppe, R. Vision-based human motion analysis: an overview. CVIU 108(1–2) (2007), 4–18.

[4] N.R. Howe, M.E. Leventon, W.T. Freeman, Bayesian reconstruction of 3D human motion from single-camera videoAdvances in Neural Information Processing Systems, vol. 12, MIT Press, Cambridge, MA, 2000.

[5] Leonid Sigal, Michael J. Black, Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion, Technical Report CS-06-08, Brown University, Department of Computer Science, Providence, RI, September 2006.

[6] Moeslund T, Granum E, A survey of computer vision-based human motion capture. Computer Vision Image Understand. 81, 3, 231–268. 2001.

[7] R. Zhu and Z. Zhou. A real-time articulated human motion tracking using tri-axis inertial/magnetic sensors package. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 12(2):295–302, 2004.

[8] Aggarwal, J.K.; Cai, Q., "Human motion analysis: a review," Nonrigid and Articulated Motion Workshop, 1997. Proceedings., IEEE , vol., no., pp.90,102, 16 Jun 1997

[9] Cai, Q.; Aggarwal, J.K., "Tracking human motion using multiple cameras," Pattern Recognition, 1996., Proceedings of the 13th International Conference on , vol.3, no., pp.68,72 vol.3, 25-29 Aug 1996

[10] R. Okada, S. Soatto, Relevant feature selection for human pose estimation and localization in cluttered images, in: Europeon Conference on Computer Vision, vol. 2, 2008, pp. 434–445.

[11] Shian-Ru Ke, Jenq-Neng Hwang, Kung-Ming Lan, Shen-Zheng Wang, "View-Invariant 3D Human Body Pose Reconstruction using a Monocular Video Camera," IEEE International Conference on Distributed Smart Camera (ICDSC), Ghent, Belgium, 2011.

[12] Baecker, R. and Buxton, W. Readings in Human-Computer Interaction - A Multidisciplinary Approach, p606, Morgan Kaufmann, Los Altos, California, 1987.

[13] Nardi, Bonnie A. Context and consciousness: activity theory and human-computer interaction. The MIT Press, 1996.

[14] Pavlovic, Vladimir I., Rajeev Sharma, and Thomas S. Huang. "Visual interpretation of hand gestures for human-computer interaction: A review."Pattern Analysis and Machine Intelligence, IEEE Transactions on 19.7 (1997): 677-695.

[15] Lee, L., R. Romano, and G. Stein. "Introduction to the special section on video surveillance." IEEE Transactions on Pattern Analysis and Machine Intelligence 22.8 (2000): 745.

[16] Hu, Weiming, et al. "A survey on visual surveillance of object motion and behaviors." Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 34.3 (2004): 334-352.

[17] Benfold, Ben, and Ian Reid. "Guiding visual surveillance by tracking human attention." British Machine Vision Conference. Vol. 459. 2009.

[18] Carranza, Joel, et al. "Free-viewpoint video of human actors." ACM Transactions on Graphics (TOG) 22.3 (2003): 569-577.

[19] Lee, Jehee, et al. "Interactive control of avatars animated with human motion data." ACM Transactions on Graphics (TOG). Vol. 21. No. 3. ACM, 2002.

[20] Wang, Liang, Weiming Hu, and Tieniu Tan. "Recent developments in human motion analysis." Pattern recognition 36.3 (2003): 585-601.

[21] Obdržálek, S., et al. "Real-time human pose detection and tracking for tele-rehabilitation in virtual reality." Studies in health technology and informatics 173 (2012): 320.

[22] Theobalt, Christian, et al. "Combining 2D feature tracking and volume reconstruction for online video-based human motion capture." International Journal of Image and Graphics 4.04 (2004): 563-583.

[23] R. T. Collins, A. J. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt, and L. Wixson, “A system for video surveillance and monitoring,” Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep., CMU-RI-TR-00-12, 2000.

[24] Haritaoglu, Ismail, David Harwood, and Larry S. Davis. "W 4 S: A real-time system for detecting and tracking people in 2 1/2D." Computer Vision—ECCV'98. Springer Berlin Heidelberg, 1998. 877-892.

[25] Ju, Shannon X., Michael J. Black, and Yaser Yacoob. "Cardboard people: A parameterized model of articulated image motion." Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on. IEEE, 1996.

[26] Sidenbladh, Hedvig, Michael J. Black, and David J. Fleet. "Stochastic tracking of 3D human figures using 2D image motion." Computer Vision—ECCV 2000. Springer Berlin Heidelberg, 2000. 702-718.

[27] Fablet, Ronan, and Michael J. Black. "Automatic detection and tracking of human motion with a view-based representation." Computer Vision—ECCV 2002. Springer Berlin Heidelberg, 2002. 476-491.

[28] Sigal, Leonid, et al. "Attractive people: Assembling loose-limbed models using non-parametric belief propagation." Advances in neural information processing systems 16 (2003).

[29] Sturm, Peter, and Bill Triggs. "A factorization based algorithm for multi-image projective structure and motion." Computer Vision—ECCV'96. Springer Berlin Heidelberg, 1996. 709-720.

[30] Sminchisescu, Cristian, and Bill Triggs. "Covariance scaled sampling for monocular 3D body tracking." Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. Vol. 1. IEEE, 2001.

[31] Agarwal, Ankur, and Bill Triggs. "3D human pose from silhouettes by relevance vector regression." Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on. Vol. 2. IEEE, 2004.

[32] Agarwal, Ankur, and Bill Triggs. "Learning to Recover 3D Human Pose from Silhouettes." Learning 2004-Abstracts of the 2004 Snowbird Learning Workshop. 2004.

[33] Sminchisescu, Cristian, and Bill Triggs. "Estimating articulated human motion with covariance scaled sampling." The International Journal of Robotics Research 22.6 (2003): 371-391.

[34] John, Vijay, Emanuele Trucco, and Spela Ivekovic. "Markerless human articulated tracking using hierarchical particle swarm optimisation." Image and Vision Computing 28.11 (2010): 1530-1547.

[35] Ivekovic, Špela, Emanuele Trucco, and Yvan R. Petillot. "Human body pose estimation with particle swarm optimisation." Evolutionary Computation 16.4 (2008): 509-528.

[36] Mussi, Luca, Spela Ivekovic, and Stefano Cagnoni. "Markerless articulated human body tracking from multi-view video with GPU-PSO." Evolvable Systems: From Biology to Hardware. Springer Berlin Heidelberg, 2010. 97-108.

[37] Ivekovic, Spela, Vijay John, and Emanuele Trucco. "Markerless multi-view articulated pose estimation using adaptive hierarchical particle swarm optimisation." Applications of Evolutionary Computation. Springer Berlin Heidelberg, 2010. 241-250.

[38] Urtasun, Raquel, David J. Fleet, and Pascal Fua. "3D people tracking with Gaussian process dynamical models." Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. Vol. 1. IEEE, 2006.

[39] Kapoor, Ashish, et al. "Gaussian processes for object categorization."International Journal of Computer Vision 88.2 (2010): 169-188.

[40] Urtasun, Raquel, David J. Fleet, and Neil D. Lawrence. "Modeling human locomotion with topologically constrained latent variable models." Human Motion–Understanding, Modeling, Capture and Animation. Springer Berlin Heidelberg, 2007. 104-118.

[41] Shotton, Jamie, et al. "Real-time human pose recognition in parts from single depth images." Communications of the ACM 56.1 (2013): 116-124.

[42] Nedel, L. Porcher. "Simulating virtual humans." Simposio brasileiro de computacao grafica e processamento de imagens, Rio de Janeiro. 1998.

[43] Huston, R. L., and C. E. Passerello. "On the dynamics of a human body model." Journal of Biomechanics 4.5 (1971): 369-378.

[44] Ning, Huazhong, et al. "Kinematics-based tracking of human walking in monocular video sequences." Image and Vision Computing 22.5 (2004): 429-441.

[45] Sand, Peter, Leonard McMillan, and Jovan Popović. "Continuous capture of skin deformation." ACM Transactions on Graphics (TOG) 22.3 (2003): 578-586.

[46] Chen, Shengyong. "Modeling for Deformable Body and Motion Analysis: A Review." Mathematical Problems in Engineering 2013 (2013).

[47] Horn, Berthold KP, and Brian G. Schunck. "Determining optical flow." Artificial intelligence 17.1 (1981): 185-203.
[48] Bobick, Aaron F., and James W. Davis. "The recognition of human movement using temporal templates." Pattern Analysis and Machine Intelligence, IEEE Transactions on 23.3 (2001): 257-267.

[49] Kennedy, James, and Russell Eberhart. "Particle swarm optimization." Neural Networks, 1995. Proceedings., IEEE International Conference on. Vol. 4. IEEE, 1995.

[50] Reynolds, Craig W. "Flocks, herds and schools: A distributed behavioral model." ACM SIGGRAPH Computer Graphics. Vol. 21. No. 4. ACM, 1987.

[51] Wilson, Edward O. "What is sociobiology?." Society 15.6 (1978): 10-14.

[52] Shi, Yuhui. "Particle swarm optimization: developments, applications and resources." Evolutionary Computation, 2001. Proceedings of the 2001 Congress on. Vol. 1. IEEE, 2001.

[53] Arthur, W. Brian, et al. Asset pricing under endogenous expectation in an artificial stock market. No. 96-12-093. 1996.

[54] Sidenbladh, Hedvig, Michael J. Black, and Leonid Sigal. "Implicit probabilistic models of human motion for synthesis and tracking." Computer Vision—ECCV 2002. Springer Berlin Heidelberg, 2002. 784-800.

[55] Hogg, David. "Model-based vision: a program to see a walking person." Image and vision computing 1.1 (1983): 5-20.

[56] Zivkovic, Zoran. "Improved adaptive Gaussian mixture model for background subtraction." Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on. Vol. 2. IEEE, 2004.

[57] Sminchisescu, Cristian, and Bill Triggs. "A robust multiple hypothesis approach to monocular human motion tracking." (2001).

[58] Tong, Minglei, Yuncai Liu, and Thomas S. Huang. "3D human model and joint parameter estimation from monocular image." Pattern recognition letters 28.7 (2007): 797-805.

[59] Sminchisescu, Cristian, and Alexandru Telea. "Human pose estimation from silhouettes. a consistent approach using distance level sets." 10th International Conference on Computer Graphics, Visualization and Computer Vision (WSCG'02). Vol. 10. 2002.

[60] Khoshelham, Kourosh. "Accuracy analysis of kinect depth data." ISPRS workshop laser scanning. Vol. 38. 2011.