Robust Video Object Tracking via Camera Self-calibration

Zheng Tang

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

Doctor of Philosophy

University of Washington

2019

Reading Committee:

Jenq-Neng Hwang, Chair

Linda G. Shapiro

Ming-Ting Sun

Program Authorized to Offer Degree:
Electrical and Computer Engineering

University of Washington

Abstract

Robust Video Object Tracking via Camera Self-calibration

Zheng Tang

Chair of the Supervisory Committee:
Professor Jenq-Neng Hwang
Department of Electrical and Computer Engineering

In this dissertation, a framework for 3D scene reconstruction based on robust video object tracking assisted by camera self-calibration is proposed, which includes several algorithmic components. (1) An algorithm for joint camera self-calibration and automatic radial distortion correction based on tracking of walking persons is designed to convert multiple object tracking into 3D space. (2) An adaptive model that learns online a relatively long-term appearance change of each target is proposed for robust 3D tracking. (3) We also develop an iterative two-step evolutionary optimization scheme to estimate 3D pose of each human target, which can jointly compute the camera trajectory for a moving camera as well. (4) With 3D tracking results and human pose information from multiple views, we propose multi-view 3D scene reconstruction based on data association with visual and semantic attributes.

Camera calibration and radial distortion correction are crucial prerequisites for 3D scene understanding. Many existing works rely on the Manhattan world assumption to estimate camera parameters automatically, however, they may perform poorly when lack of man-made structure in the scene. As walking humans are common objects in video analytics, they have also been used for camera calibration, but the main challenges include noise reduction for the estimation of vanishing points, the relaxation of assumptions on unknown camera parameters, and radial distortion correction. We propose a novel framework for camera self-calibration and automatic radial distortion correction. Our approach starts with

a multi-kernel-based adaptive segmentation and tracking scheme that dynamically controls the decision thresholds of background subtraction and shadow removal around the adaptive kernel regions based on the preliminary tracking results. With the head/foot points collected from tracking and segmentation results, mean shift clustering and Laplace linear regression are introduced in the estimation of the vertical vanishing point and the horizon line, respectively. The estimation of distribution algorithm (EDA), an evolutionary optimization scheme, is then utilized to optimize the camera parameters and distortion coefficients, in which all the unknowns in camera projection can be fine-tuned simultaneously. Experiments on three public benchmarks and our own captured dataset demonstrate the robustness of the proposed method. The superiority of this algorithm is also verified by the capability of reliably converting 2D object tracking into 3D space.

Multiple object tracking has been a challenging field, mainly due to noisy detection sets and identity switch caused by occlusion and similar appearance among nearby targets. Previous works rely on appearance models built on individual or several selected frames for the comparison of features, but they cannot encode long-term appearance change caused by pose, viewing angle and lighting condition. We propose an adaptive model that learns online a relatively long-term appearance change of each target. The proposed model is compatible with any features of fixed dimension or their combinations, whose learning rates are dynamically controlled by adaptive update and spatial weighting schemes. To handle occlusion and nearby objects sharing similar appearance, we also design cross-matching and re-identification schemes based on the proposed adaptive appearance models. Additionally, the 3D geometry information is effectively incorporated in our formulation for data association. The proposed method outperforms all the state-of-the-art on the *MOTChallenge* 3D benchmark and achieves real-time computation with only a standard desktop CPU. It has also shown superior performance over the state-of-the-art on the 2D benchmark of *MOTChallenge*.

For more comprehensive 3D scene reconstruction, we develop a monocular 3D human pose estimation algorithm based on two-step EDA that can simultaneously estimate the camera motion for a moving camera. We first derive reliable 2D joint points through deep-learning-based 2D pose estimation and feature tracking. If the camera is moving, the initial camera poses can be estimated from visual odometry, where the feature points extracted on the human bodies are removed by segmentation masks dilated from 2D skeletons. Then the 3D joint points and camera parameters are iteratively optimized through a two-step evolutionary algorithm. The cost function for human pose optimization consists of loss terms defined by spatial and temporal constancy, "flatness" of human bodies, and joint angle constraints. On the other hand, the optimization for camera movement is based on the minimization of reprojection error of skeleton joint points. Extensive experiments have been conducted on various video data, which verify the robustness of the proposed method.

The final goal of our work is to fully understand and reconstruct the 3D scene, *i.e.*, to recover the trajectory and action of each object. The above methods can be extended to a system with camera array of overlapping views. We propose a novel video scene reconstruction framework to collaboratively track multiple human objects and estimate their 3D poses across multiple camera views. First, tracklets are extracted from each single view following the tracking-by-detection paradigm. We propose an effective integration of visual and semantic object attributes, including appearance models, geometry information and poses/actions, to associate tracklets across different views. Based on the optimum viewing perspectives derived from tracking, we generate the 3D skeleton of each object. The estimated body joint points are fed back to the tracking stage to enhance tracklet association. Experiments on a benchmark of multi-view tracking validate our effectiveness.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

CAMERA CALIBRATION: the process of estimating intrinsic and/or extrinsic camera parameters. Intrinsic parameters deal with the camera's internal characteristics, such as, its focal length, skew, and image center. Extrinsic parameters describe its position and orientation in the world.

CAMERA COORDINATE SYSTEM (CCS): 3D Cartesian coordinate system where the camera is located at the origin.

CAMERA SELF-CALIBRATION: the process of determining internal camera parameters directly from multiple uncalibrated images of unstructured scenes.

CONVOLUTIONAL NEURAL NETWORK (CNN): a class of deep neural networks, most commonly applied to analyzing visual imagery. CNNs are regularized versions of multilayer perceptrons. Multilayer perceptrons usually refer to fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer.

ESTIMATION OF DISTRIBUTION ALGORITHM (EDA): stochastic optimization method that guides the search for the optimum by building and sampling explicit probabilistic models of promising candidate solutions. Optimization is viewed as a series of incremental updates of a probabilistic model, starting with the model encoding the uniform distribution over admissible solutions and ending with the model that generates only the global optima. EDAs belong to the class of EAs.

EVOLUTIONARY ALGORITHM (EA): a subset of evolutionary computation, a generic population-based metaheuristic optimization algorithm. An EA uses mechanisms inspired by biological evolution, such as reproduction, mutation, recombination, and selection.

FIELD OF VIEW (FOV): the angle through which the devices can pick up electromagnetic radiation. FOV allows for coverage of an area rather than a single focused point.

HORIZON LINE (L_∞): the 2D line that represents the extension of the 3D ground plane to infinity on the image plane.

IMAGE SEGMENTATION: the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics, *e.g.*, belong to the same object.

KALMAN FILTERING: an algorithm that uses a series of measurements observed over time, containing statistical noise and other inaccuracies, and produces estimates of unknown variables that tend to be more accurate than those based on a single measurement alone, by estimating a joint probability distribution over the variables for each time frame.

LEVENBERG–MARQUARDT (LM) ALGORITHM: an approach also known as the damped least-squares (DLS) method, that leverages gradient descent to solve non-linear least squares problems.

MANHATTAN WORLD ASSUMPTION (MWA): the assumption that states city and indoor scenes are built on a Cartesian grid which leads to regularities in the image edge gradient statistics.

MOSTLY LOST TARGETS (ML): a measure of MOT performance that gives the ratio of ground-truth trajectories that are covered by a track hypothesis for at most 20% of their respective life span.

MULTIPLE OBJECT TRACKING (MOT): tracking the locations of multiple targets in a video sequence.

MULTIPLE OBJECT TRACKING ACCURACY (MOTA): a measure of MOT performance combining three error sources: false positives, missed targets and identity switches.

MULTIPLE OBJECT TRACKING PRECISION (MOTP): a measure of MOT performance based on the misalignment between the annotated and the predicted bounding boxes.

MOSTLY TRACKED TARGETS (MT): a measure of MOT performance that gives the ratio of ground-truth trajectories that are covered by a track hypothesis for at least 80% of their respective life span.

OBJECT DETECTION: detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos.

OPTICAL FLOW: the pattern of apparent motion of image objects between two consecutive frames caused by the movemement of object or camera. It is 2D vector field where

each vector is a displacement vector showing the movement of points from first frame to second.

PINHOLE CAMERA MODEL: the mathematical relationship between the coordinates of a point in 3D space and its projection onto the image plane of an ideal pinhole camera.

POSE ESTIMATION: detection of the position and orientation of an object. This usually means detecting keypoint locations that describe the object.

PRINCIPAL COMPONENT ANALYSIS (PCA): a statistical procedure that extracts the most important features of a dataset.

REGION OF INTEREST (ROI): a portion of an image that we want to filter or perform some other operation on.

TRACKING BY DETECTION: a tracking paradigm defined by association of object detection results in time.

TRACKING BY SEGMENTATION: a tracking paradigm defined by association of object segmentation results in time.

TRACKLET: a series of human bounding boxes grouped by spatio-temporal coherency and perceptual similarity.

RADIAL DISTORTION: distortions that are radially symmetric, or approximately so, arising from the symmetry of a photographic lens. This kind of distortion appears most visibly when the widest angle (shortest focal length) is selected either with a fixed or a zoom lens.

RANDOM SAMPLE CONSENSUS (RANSAC): an iterative method to estimate parameters of a mathematical model from a set of observed data that contains outliers, when outliers are to be accorded no influence on the values of the estimates.

RECURRENT NEURAL NETWORK (RNN): a class of artificial neural network where connections between nodes form a directed graph along a temporal sequence.

REPROJECTION ERROR: a geometric error corresponding to the image distance between a projected point and a measured one.

SCENE RECONSTRUCTION: the process of capturing the location and shape of each real object. This process can be accomplished either by active or passive methods.

VANISHING POINT: the point at which receding parallel lines viewed in perspective appear to converge.

VERTICAL VANISHING POINT (V_∞): the vanishing point that does not locate on the horizon line, which is formed by parallel lines that are perpendicular to the ground plane.

VIDEO ANALYTICS: the capability of automatically analyzing video to detect and determine temporal and spatial events.

VISUAL OBJECT TRACKING (VOT): the estimation of the location of a target in all the frames of a video sequence based on the given initial location (or a bounding rectangle) of the target.

VISUAL ODOMETRY: estimation of the motion of a camera in real time using sequential images (*i.e.*, egomotion).

WORLD COORDINATE SYSTEM (WCS): 3D Cartesian coordinate system where the origin is pre-defined and does not change with camera movement.

ACKNOWLEDGMENTS

I owe many people a debt of gratitude for all their support over the years in graduate school.

First, I am indebted to my advisor, Prof. Jenq-Neng Hwang. Since my first day in the doctoral program, Prof. Hwang believed in me like nobody else and gave me endless support. Without the generous offer he extended to me and all the research assistantships, I will never have chance to complete this dissertation. On the academic level, Prof. Hwang taught me how to define a research problem, find a solutioin to it, and finally publish the results. On a personal level, Prof. Hwang inspired me by his hardworking and passionate attitude.

Besides my advisor, I would like to give thanks to my dissertation defense committee, including Prof. Linda G. Shapiro, Prof. Ming-Ting Sun, Prof. Fa-Long Luo, and Prof. Kenneth P. Bube, for their invaluable guidance and advice. I am thankful to Prof. Shapiro for her insightful comments and for sharing her tremendous experience in the computer vision field. I am also grateful to Prof. Sun, an expert in video processing, computer vision and machine learning, for his crucial remarks that shaped my final dissetation. I am sincerely appreciative of Prof. Fa-Long Luo, who referred me to this program and continued to support me in my research. Last but not least, I also show gratitude to Prof. Bube, an excellent teacher and a friend, who gave me enormous encouragement to pursue my doctoral degree.

I also would like to thank my lab mates for their continued support. This dissertation would not have been possible without the collaboration with many of them in various projects related to this dissertation. Especially, I am grateful to all the team members at the 2017 AI City Challenge and the 2018 AI City Challenge. It has been my pleasure to lead this winning team in both years, and bring honor to the Department of Electrical and Computer

Engineering and the University of Washington.

Besides, I am also grateful to my industrial collaborators. I spent nine months on an internship at NVIDIA where I had the chance to collaborate with fantastic researchers. More specifically, I would like to thank Dr. Milind Naphade and Dr. Stan Birchfield for their mentorship and for providing me the great opportunity to work on the 2019 AI City Challenge. I also extend my gratitude to many companies and individuals for their financial support that I otherwise would not have been able to develop my scientific discoveries, including Madrona Venture Labs, Prism Skylabs, Mr. Wanhai Cui, ArchieMD Inc., *etc.*

Finally, but by no means least, I would like to express my deepest gratitude to my family and friends. In particular, I would like to thank my parents who raised me with warm love and continued patience, and supported me in all my pursuits. I am also grateful for Yijin Lee's faithful support during the final stages of this doctoral degree. This last word of acknowledgment I have saved for my brothers and sisters at the University Presbyterian Church, who have made Seattle a home away from home for me.

DEDICATION

to my Lord and Savior, Jesus Christ

Chapter 1

INTRODUCTION

The growing demand of user experience with video streaming has brought about a rapid growth in big visual data analytics. The ultimate purpose of major applications in this research field is to fully understand and reconstruct the video scene in a 3D space. This not only involves accurate identification of multiple objects and the recovery of their trajectories, but also requires precise estimation of their postures (see figure 1.1). The three main components in 3D scene reconstruction are (1) object tracking, (2) camera calibration, and (3) pose estimation.

1.1 Object Tracking

Object tracking is a key issue in many computer vision applications. The task of single-target tracking (a.k.a. *visual object tracking* (VOT)) has been well addressed by leveraging the cues of appearance or silhouette of selected target. On the other hand, the goal of *multiple object tracking* (MOT) is to simultaneously recover the trajectories of many targets of interest in a video sequence. Though MOT has seen considerable progress in recent years because of improved appearance models and optimization schemes, the status quo is still far from matching human performance, due to the difficult data association problem and the interactions among objects. Furthermore, because of variations in number of targets, we need to design schemes to effectively initialize object locations in every frame, which are usually derived from appearance-based object detection or background subtraction. The former category is called *tracking by detection*, and the latter *tracking by segmentation*.

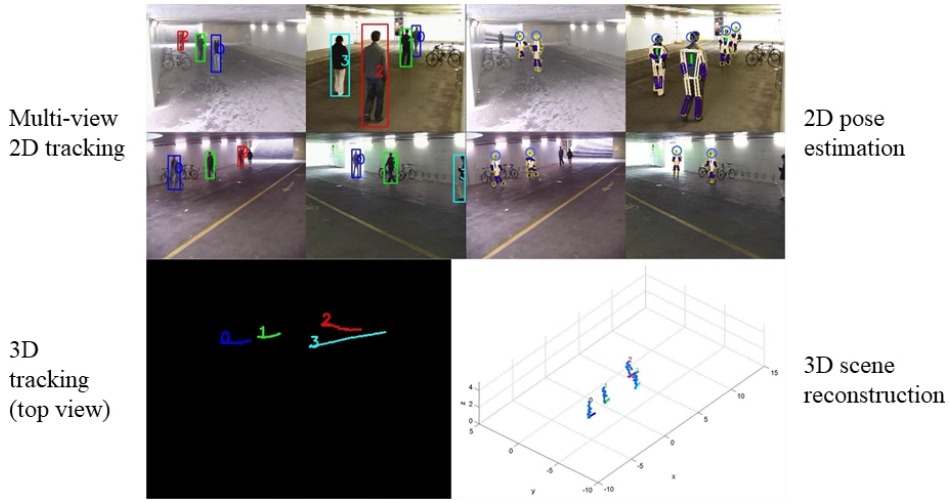

Figure 1.1: This is a demonstration of the stages in 3D scene reconstruction on the EPFL dataset[4]. Note that the proposed framework can be applied to a single camera view, or extended to multiple cameras with overlapping view. First, 2D tracking is conducted in each single view (top left). From the tracking of walking humans, camera self-calibration is applied for computing camera parameters, which can convert tracking into 3D space (bottom left). Similarly, 2D pose estimation is processed in each single view (top right) and projected into 3D. When 3D pose is combined with the 3D tracking information, the scene is reconstructed through data association across camera views (bottom right).

1.1.1 Tracking by Detection

Most state-of-the-art[51, 2, 78, 129] are in the tracking-by-detection school, in which they exploit the continuity in space and time, but the information of object appearance is seldom considered to facilitate tracking. The major challenges of tracking by detection include noise in object detection, appearance change, and identity switch caused by object occlusion and similar appearance between objects in pair/group.

Most of the state-of-the-art methods in tracking by detection focus on data association techniques. The majority of them are offline algorithms, *e.g.*, [51, 129, 128], in which obser-

vations of objects are grouped into tracklets based on spatio-temporal continuity. In data association, besides motion patterns and social force models, appearance models have also been widely used as an important cue to keep the identities of targets. Traditionally, appearance models based on raw pixel template representation[92, 89], fusion of color/texture/edge features[146], or color/texture/edge histograms[128, 18, 17, 52] are adopted for their simplicity. Nevertheless, these models are only built on individual or several selected frames, which could not encode long-term appearance change along each trajectory. Thus, they may fail when there is change of lighting condition, viewing angle or object pose. Other researchers also introduce methods based on random forest algorithms[44, 45] or take advantage of deep learning features[100] to improve the robustness of appearance modeling, but the computation complexity significantly increases and massive training samples are required.

Inspired by adaptive background modeling in change detection[3, 37, 105, 106], we propose an adaptive appearance model that can learn the long-term change of object appearance online. The proposed framework, termed MOANA which is short for "Modeling of Object Appearance by Normalized Adaptation", models the appearance of each target as a normalized matrix with an array of observed feature vectors at each cell. MOANA is compatible with any features of fixed dimension or their combination. To update the model, the learning rates are controlled by the similarity with previous features and spatial weighting. When an object is partially occluded by or spatially close to others, a cross-matching module is employed to avoid identity switch based on the proposed appearance model. For objects that are seriously occluded or failed to be detected (false negatives) for a few frames, we design a re-identification scheme to recover their trajectories. 3D geometry information is also leveraged in our formulation of data association. Experiments are conducted on the test and training sets of the *MOTChallenge* 3D benchmark[50]. We are ranked on top of the benchmark in terms of the multiple object tracking accuracy (MOTA)[5]. Our proposed method has also shown superior performance over the state-of-the-art in the *MOTChallenge* 2D benchmark[50]. A demonstration of 3D tracking and visualization of averaged adaptive appearance features are presented in figure 1.2.

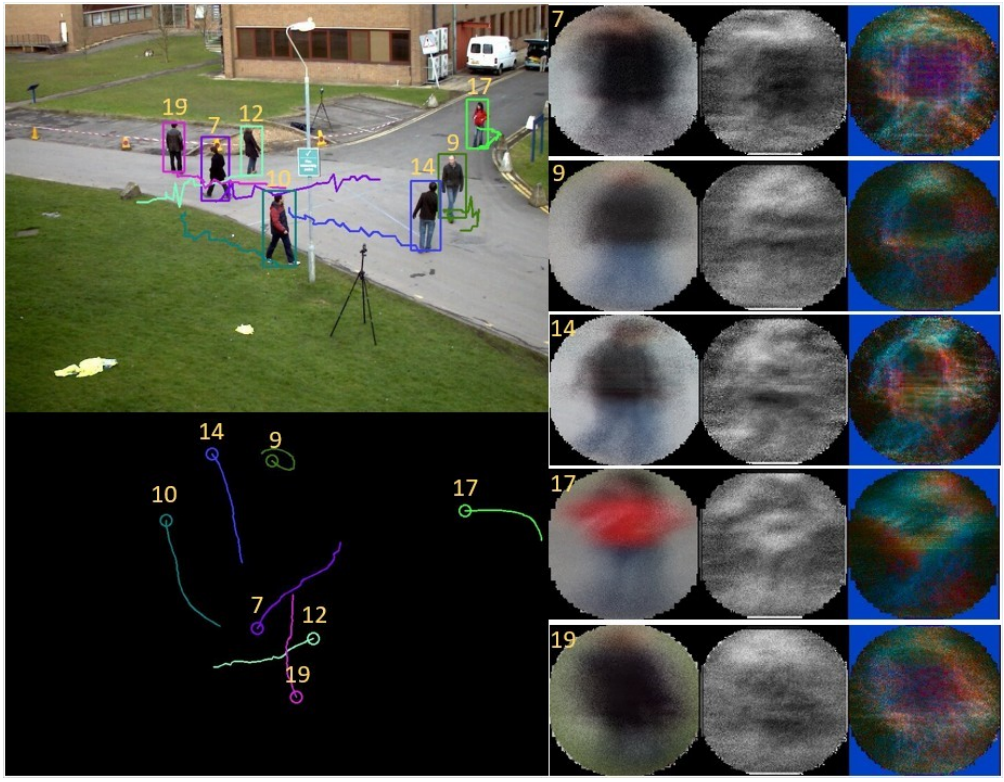

Figure 1.2: This figure shows MOT in 2D (top-left), the back projection to 3D in top view (bottom-left) and the visualization of the averaged adaptive appearance models learned online in RGB space, *local binary patterns* (LBP) space, and gradient space (right).

1.1.2 Tracking by Segmentation

For tracking by segmentation, the failure in foreground object segmentation can severely degrade the performance of tracking. It usually occurs in the following scenario: when an object enters into a camera's field of view in which some parts of the body have similar color with the modeled background, object segmentation can easily fail because the corresponding foreground regions are likely to be merged into background or recognized as shadow (the problem of *object merging*). These will eventually lead to a dead loop that results in increasing errors in tracking.

To acquire robust tracking and segmentation results to support camera self-calibration, we propose an algorithm to deal with object merging, named *Multi-kernel Adaptive Segmentation and Tracking* (MAST). A feedback loop originated from preliminary tracking results is used to help preserve segmented foreground when object(s) share color or chromaticity similarity with background. The region-level feedback can be derived from the kernel histograms built in *constrained multi-kernel* (CMK)[17] tracking. We also introduce effective penalty computation for shadow removal based on the distance between chromaticity histograms of foreground and background.

1.1.3 Cross-view Tracking

Recently, cross-view tracking of multiple people in a camera array has attracted lots of attentions in the literature[69]. Researchers exploit multiple cues in both 2D and 3D, *e.g.*, ground plane occupancy[4, 29, 62], motion coherence, appearance affinity[133], temporal consistency[66], postures and actions[134], *etc.*, to locate multiple targets in a 3D scene map. Nonetheless, there remain many challenges that have not been fully resolved. First, in crowded scenes where people frequently occlude and intersect with one another, the number of identity switches can increase rapidly. Moreover, the same object may experience large appearance variation across different viewpoints. Last but not least, the inaccuracy of ground plane estimation causes mistakes in geo-localization, especially for objects that are far away in a video scene.

In our framework of 3D scene reconstruction, we propose a multi-view multi-object tracking method based on energy minimization. For initialization, we follow the tracking-by-detection paradigm to generate *tracklets* in each single view, which are series of human bounding boxes grouped by spatio-temporal coherency and perceptual similarity. Then we formulate the data association problem as energy minimization based on a set of visual and semantic attributes, including a pixel-based adaptive model for two-way appearance comparison and a geometry proximity measurement based on weighting of depth and visibility. We also introduce an explicit action descriptor using feedback from the pose estimation stage.

1.2 Camera Calibration

The problems in MOT can be mitigated when the correspondence between the 2D image plane and the 3D space in real world is established. Thus, the tracking space can be converted into 3D, where depth information can be effectively utilized, and the prediction of object movement and scale can be more reliable. Most existing works adopt the *pinhole camera model* to compute the 3D-to-2D projection relationship, *i.e.*, *camera calibration*. The camera parameters for projection consist of *intrinsic parameters*, which encode the *camera coordinate system* (CCS), and *extrinsic parameters*, which describe the transformation to the *world coordinate system* (WCS). Sometimes, the camera may also suffer from *radial distortion*, manifested in form of the "fish-eye" effect. The computation of camera parameters and distortion coefficients can be formulated as a *Perspective-n-Point* problem when sufficient measurements of 3D points are available, which may be derived from some calibration templates. However, these manual solutions require time-consuming annotation and interaction at the scene, which make them infeasible for a large-scale camera network. Moreover, for the widely installed *pan-tilt-zoom* (PTZ) cameras, the camera parameters may change occasionally that makes the previous measurements invalid. Therefore, many approaches have been proposed to automatically calibrate the cameras based on assumptions on the camera scenes. This category of methods is termed as *camera self-calibration*.

Most methods in camera self-calibration try to find the *vanishing points* of parallel lines in the 3D real world. Caprile and Torre[11] first propose to recover both intrinsic and extrinsic parameters from given vanishing points. Later, many works[58, 25], based on the *Manhattan world assumption* (MWA), utilize vanishing points from regular architectural structures in the scene for camera calibration. However, MWA is invalid for many scenarios, where the observation of common video objects, *e.g.*, pedestrians and vehicles, can thus be utilized for camera self-calibration.

In [70], Lv *et al.* propose a method for camera self-calibration from observation of a human walking on a planar surface. Each human instance can be modeled as a vertical pole

with constant height that is perpendicular to the ground plane, from which they calculate the *vertical vanishing point*, V_∞, and the *horizon line*, L_∞. Then the camera parameters can be computed based on some assumptions on the intrinsic camera parameters. Though many other algorithms[71, 46, 42, 132, 48, 63, 38, 9, 31, 112] have been developed to improve their performance, this task is still facing a few challenges. First, Mohedano and Garcia[79] analyze the limitation of single-camera-based self-calibration from human tracking, from which they conclude that this formulation is not applicable for a camera with unknown aspect ratio of focal lengths, principal point coordinates and skew. In other words, to apply this method, we need to assume that the focal length is the only unknown intrinsic camera parameter to be estimated. The ambiguity caused by such assumptions leads to the increase of *reprojection error*. The second challenge lies in noise reduction for the estimation of V_∞ and L_∞. The noise and outliers are mainly caused by the uncertainty in head/foot localization. Among the previous works, RANSAC has been the most popular approach adopted[71, 132, 63, 31]. Unfortunately, in most scenarios where the number of outliers overwhelms inliers, the performance of RANSAC degrades. Additionally, the threshold to indicate inliers in RANSAC needs to be fine-tuned for different scenarios. Last but not least, all the previous methods cannot be applied to a severely distorted camera, such as a wide-angle or fish-eye camera, which requires additional estimation of distortion coefficients.

We propose a novel framework for joint camera self-calibration and automatic radial distortion correction from the tracking of walking humans. It is entitled ESTHER, short for "Evolutionary Self-calibration from Tracking of Humans for Enhancing Robustness". To the best of our knowledge, ESTHER is the first work on video-object-based automatic recovery from radial distortion. In brief, we first collect head/foot points of walking humans based on adaptive segmentation and tracking. Mean shift clustering and Laplace linear regression are respectively employed in the estimation of V_∞ and L_∞ to overcome the deficiencies of RANSAC. To relax the assumptions on unknown intrinsic camera parameters, we take advantage of the evolutionary algorithm to optimize camera parameters. The final step is to correct radial distortion, which also exploits evolutionary optimization to search for

the optimal distortion coefficients. The optimization of camera parameters and distortion coefficients iterates until the stopping criterion is met.

1.3 Pose Estimation

Human pose estimation is a fundamental yet challenging problem in computer vision. The goal is to estimate 2D or 3D locations of body parts given an image or a video, which provides informative knowledge for tasks such as action recognition, robotics vision, human-computer interaction, and autonomous driving. Significant advances have been achieved in 2D human pose estimation recently because of the powerful *convolutional neural networks* (CNNs) and the availability of large-scale in-the-wild human pose datasets with manual annotations. However, advances in 3D human pose estimation remain limited. The major challenge is the under-constrained nature of the problem due to loss of depth information and frequent (self-)occlusion.

Existing datasets such as *Human3.6M* [41, 40] are collected in constrained lab environment using mocap systems, hence the variations in background, viewpoint, and lighting are very limited. Although CNNs fit well on these datasets, when being applied on in-the-wild images, where only 2D ground-truth annotations are available (*e.g.*, the MPII human pose dataset[1]), they may have difficulty in terms of generalization ability due to the large domain shift between constrained lab environment images and unconstrained in-the-wild images. This task gets especially challenging when the target videos are under high compression, due to limited wireless bandwidth in network transmission.

In our scenario, the goal is to develop an anatomy-enabled augmented reality, point-of-care, telemedicine application. The estimated 3D human pose is employed to automatically locate the human joints and enable efficient overlaying virtual anatomy, as demonstrated in figure 1.3. The proposed system transmits the compressed video from the mobile phone of the combat medic to the physician's laptop for remote processing, and then the overlaid anatomy is transmitted back to the combat medic's device. The complete work flow is shown in figure 1.4. The task to be addressed in this dissertation is the 3D pose estimation

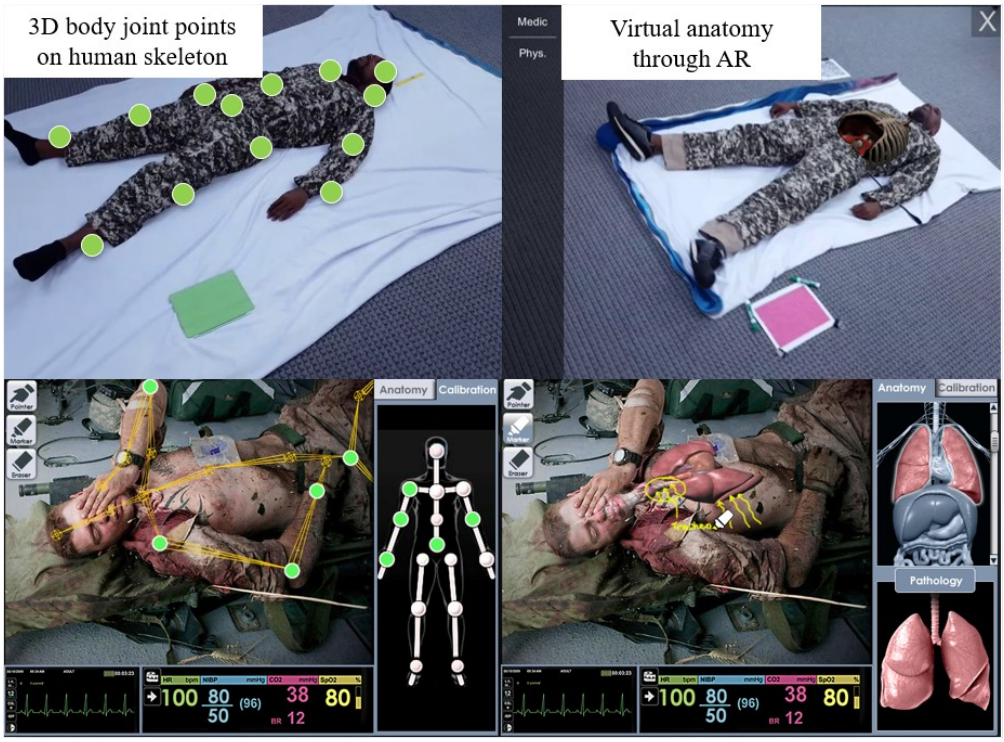

Figure 1.3: This is a demonstration of virtual anatomy in the telemedicine application. On the top of the figure, we show the overlaid 3D body joint points and virtual anatomy on the captured images, respectively. At the bottom, the corresponding user interfaces on the remote physician's device are displayed.

for anatomy calibration. The camera pose can also be jointly computed during the 3D human pose estimation.

For the joint estimation of camera pose and 3D human pose, we propose a two-step evolutionary algorithm, named 2EPOCH which is short for "Two-step Evolutionary Pose Optimization for Camera and Humans". More specifically, the input video frames are first processed by CNN-based 2D pose estimation. The joint points that are missing or falsely located are corrected by optical-flow-based tracking. The refined joint pairs are dilated to form masks, which will be used to remove feature points on the human bodies in visual

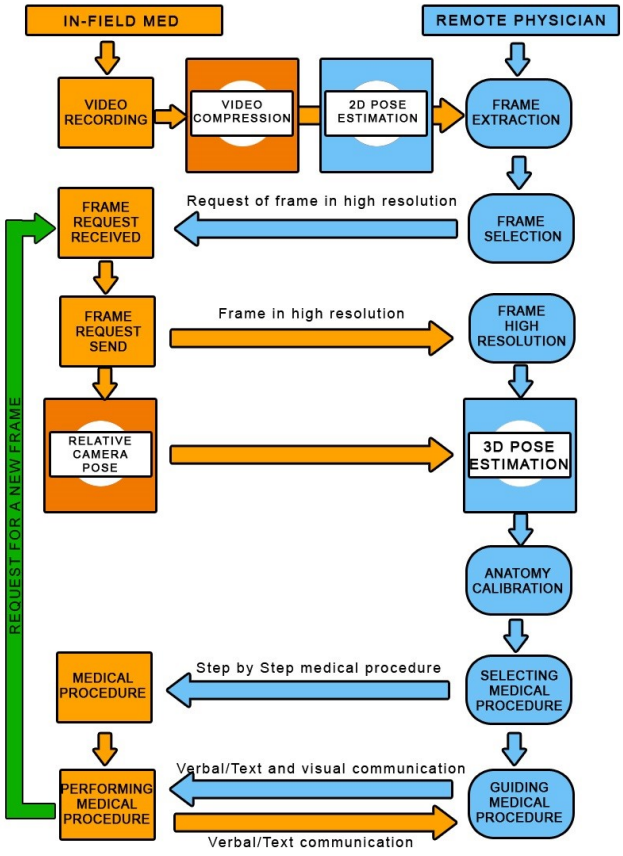

Figure 1.4: This is the flow diagram of the telemedicine application, where our task is to estimate the camera pose and 3D human pose for overlaying virtual anatomy.

odometry. Then the 2D joint points and camera position derived from visual odometry are fed into a two-step optimization module, which can iteratively search for the optimal camera pose and 3D human pose based on pre-defined cost functions.

1.4 Scene Reconstruction

The flow diagram of the proposed framework for 3D scene reconstruction is presented in figure 1.5. As demonstrated in figure 1.1, 3D scene reconstruction is the joint effort of all the algorithmic components mentioned in the previous sections. The input can be video frames

from a single camera, or multiple viewpoints captured by a static camera array sharing overlapping view. The tracking and segmentation results of walking humans from MAST are utilized for camera self-calibration based on the proposed ESTHER algorithm. With the camera projection matrix and detected objects as input, we introduce MOANA for tracking by detection in 3D. Meanwhile, the input frames are processed by 2D pose estimation, and further transformed to 3D space leveraging our 2EPOCH scheme. Note that 2EPOCH can also be applied to a moving camera and simultaneously optimize the camera pose. Finally, the 3D human joint points and tracking information from multiple viewing perspectives are effectively combined through data association formulated as energy minimization. The geometry information in multi-view tracking is used to choose the optimum viewpoint for 3D pose estimation. On the other hand, the multi-view data association in 3D tracking leverages feedback from pose estimation, which is combined with other visual and semantic attributes. Thus, the two modules can mutually benefit from each other and improve the overall performance. The output of our framework is the recovered 3D scene represented by human locations and their postures.

The work in this dissertation has been partially described in our previous publications[110, 112, 114, 65, 55, 54, 115, 125, 108, 111, 109, 113, 83]. The main contributions are summarized as follows:

- The region-level multi-kernel feedback used to enhance robustness in segmentation, and improve tracking by segmentation correspondingly;

- Two formulations based on EDA to optimize camera parameters and radial distortion coefficients that minimize reprojection error on the ground plane and the relative human height variance, respectively;

- Introduction of mean shift clustering and Laplace linear regression to vanishing points estimation for reducing noise and removing outliers;

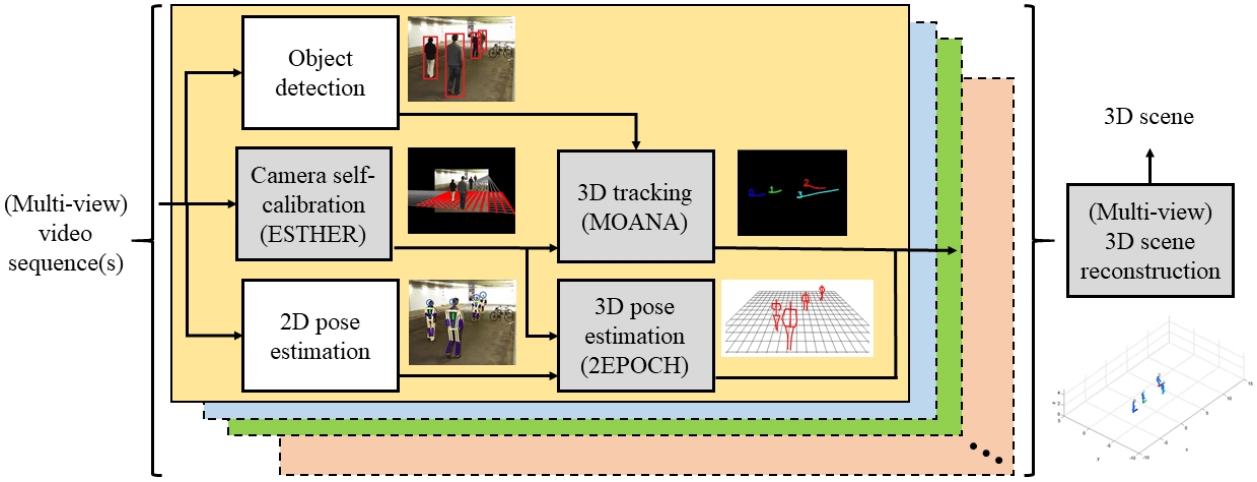

Figure 1.5: This is the flow diagram of the proposed framework for 3D scene reconstruction (with demonstration on the EPFL dataset[4] provided), which can be applied to a single view or multiple cameras sharing overlapping view. The shaded cells represent the proposed algorithmic components for this framework. Colored rectangles in the background indicate different camera views that share the same processing architecture.

- An adaptive model that encodes long-term appearance change for robust object tracking, which is inspired from adaptive background modeling in change detection;
- Cross-matching and re-identification schemes designed to overcome occlusion and ambiguity among neighboring objects, which incorporate both the adaptive appearance model and 3D geometry information;
- Two-step evolutionary optimization for joint camera pose estimation and human pose estimation in 3D;
- Novel representations of visual and semantic attributes adopted in multi-view data association, which is formulated as an energy minimization problem.

Chapter 2
RELATED WORKS

In this chapter, we list the works related to our proposed algorithms. Some of them will be compared with out methods in the experiments.

2.1 Object Tracking

2.1.1 Single-target Tracking

Among many techniques of tracking a single target, kernel-based object tracking such as mean shift tracker[22] that searches for similar candidate model around local neighboring regions, has gained high popularity, because of its fast convergence and low computation. To improve kernel-based tracking, Chu *et al.*[17] propose to handle occlusion based on adaptive multiple kernels with constraints on their spatial relation, *i.e.*, CMK tracking, and the accuracy is comparable to the state-of-the-art trackers. They also embed CMK tracking into a Kalman filtering tracking system[18] to further increase computational efficiency. To extend these methods to MOT, it is necessary to find a way to automatically define the locations of targets. Most of the top-ranked methods on the benchmark of MOT[75] depend on object detection for target initialization, but only few of them[76] consider to combine the information from segmentation to jointly improve performance.

2.1.2 Tracking by Detection

In tracking by detection, one of the traditional approaches to MOT is to predict the states, *i.e.*, location and size, of tracked targets based on Bayesian inference methods, *e.g.*, Kalman filter or particle filter[92, 18, 110]. These methods usually can achieve acceptable performance in short term, however, they tend to fail when objects are interacting with each other, *i.e.*,

under occlusion and/or movement in groups. Many recent works formulate MOT as a data association problem. Leal-Taixé *et al.*[51] propose to formulate data association by social force and grouping behavior. The probability hypothesis density (PHD) filter[129] is introduced in the formulation of multi-target state estimation for offline decision on data association. Wen *et al.*[128] uses a space-time-view hyper-graph to encode higher-order constraints in 3D. More recently, some researchers apply deep learning architectures like *recurrent neural networks* (RNNs) to the modeling of nonlinear behaviors in data association[100, 77].

Relatively little attention has been given to the development of discriminative appearance models for MOT. Methods like [92, 89, 146, 128, 17, 18, 52, 110] employ raw pixel template representation or fusion of traditional image features from a single frame to model the object appearance. As mentioned, the histogram representation is improved by Chu *et al.*[17], who build multiple spatially weighted kernel histograms with binding constraints for each target to overcome partial occlusion. Similarly, Yang and Nevatia[136] introduce *discriminative part-based appearance models* (DPAMs), which use a human part model to extract the discriminative features from unoccluded object area. However, all the mentioned appearance models are highly sensitive to the quality of the selected frame(s), which may fail occasionally due to illumination or other conditions. Besides, their similarity measurements either not or only implicitly encode the spatial distribution of appearance features. On the other hand, Kuo *et al.*[47] present *online learned discriminative appearance models* (OLDAMs) to learn the discriminative features from training samples collected online with some spatio-temporal constraints. In [135], the *conditional random field* (CRF) model is exploited to combine OLDAMs with non-linear motion patterns. Though the affinity measurement can be learned online for [47] and [135], they still only consider features extracted from single frame such as RGB color histogram and *histogram of oriented gradients* (HOG). These features can lead to tracking errors after objects being occluded for long time. There are other attempts to adapt the classifier to the changing appearance of each target by using variants of random forests[44, 45] and boosting[7]. Some also propose to apply deep learning features[100] generated from the CNNs to improve tracking performance. However, because

of the increased complexity of these methods and the potentially large number of targets, the computation requirement becomes a major challenge. Moreover, these methods require massive training samples to achieve robust performance.

2.1.3 Tracking by Segmentation

Robust object segmentation is essential for supporting tracking by segmentation. Adaptive background modeling is a key element of modern change detection algorithms. A regularized background adaptation for automatically controlling the learning rate of *Gaussian mixture model* (GMM) is presented by Lin *et al.*[59]. Barnich *et al.*[3] introduce the *visual background extractor* (ViBe) that builds the background model with a set of observed values in the past at each pixel location. Hoffmann *et al.*[37] propose the *pixel-based adaptive segmenter* (PBAS), which improves the pixel-based background modeling scheme by applying a random observation replacement policy. *Self-balanced sensitivity segmenter* (SuBSENSE)[105, 106], as an improvement to PBAS, further improves the update scheme using pixel-level feedback loops that dynamically adjust the internal configuration parameters, which allows it to rank among the top on the mainstream benchmark of change detection, CDnet[32]. Nevertheless, none of the algorithms are designed specifically for supporting tracking, as they can easily fail when target(s) encounter the problem of object merging, where the subsequent tracking stage will be negatively affected as well. Furthermore, the threshold decision mechanisms used in these methods are all in pixel level that only consider a limited neighboring region. In addition, our proposed MAST is the only one taking into account the adaptation for shadow removal, since shadow is also a key factor in object segmentation. Finally, to the best of our knowledge, the proposed adaptive appearance model, *i.e.*, MOANA, is the first to extend adaptive modeling and random update scheme in change detection to support robust object tracking. We also design cross-matching and re-identification schemes to resolve ambiguity among objects using the adaptive appearance models.

2.1.4 Cross-view Tracking

Similar to MOT in a single camera, multi-view object tracking is often formulated as data association across cameras. Berclaz *et al.*[4] and Fleuret *et al.*[29] follow a tracking-by-segmentation strategy to detect candidate targets. They respectively develop their data association approaches based on the *k-shortest paths* algorithm and the *hidden Markov process*. Xu *et al.*[133] use tracking by detection and exploit multiple cues in their hierarchical composition model. Their appearance coherence is measured by CNN features, whereas the motion information is encoded in a continuous function. Liu[66] uses raw pixel template in appearance modeling and combine it with 3D localization, spline fitting and temporal consistency in the objective. Both appearance models in [133] and [66] cannot adaptively "memorize" past feature values. Furthermore, Xu *et al.*[134] first introduce pose/action attributes in cross-view association. However, their human poses are learned from CNN features for categorization without explicit pose estimation, which may cause errors in transitions of actions.

2.2 Camera Calibration and Radial Distortion Correction

2.2.1 Camera Self-calibration from Walking Persons

Many related algorithms have been developed based on the method proposed in [70]. More specifically, Lv *et al.*[71] improve their own work by applying RANSAC in vanishing points estimation. They also optimize camera parameters based on the *Levenberg-Marquardt* (LM) algorithm. Krahnstoever and Mendonca[46] exploit Bayesian estimation for noise reduction. Junejo and Foroosh[42] adopt a different formulation based on two decomposed foot-to-head harmonic homologies, in which outliers are removed using the *truncated quadratic function*. Wu *et al.*[132] also apply RANSAC to the estimation of vanishing points from input head and foot locations. Kusakunniran *et al.*[48] introduce direct computation of projection matrix without decomposition into physical parameters. Liu *et al.*[63] present a new framework for optimizing camera parameters, such that the predicted relative human height distribution

matches with the prior knowledge. Recently, Huang *et al.*[38] develop a novel scheme that detects the image points of toes on the ground plane, which can directly infer the two vanishing points on L_∞. The work[9] proposes pre- and post-processing stages to improve the estimation of V_∞ and L_∞. Führ *et al.*[31] adopt a nonlinear cost function aiming to mostly align the orientation of the reprojected poles. In our previous work[112], the cost function to be minimized is designed as the reprojection error on the ground plane and we utilize evolutionary optimization to simultaneously optimize all the camera parameters.

Despite the improvement of these methods in noise reduction, there are still many difficulties to be addressed. First, as concluded by Mohedano and Garcia[79], the estimation of V_∞ and L_∞ depends on the unrealistic assumptions of fixed aspect ratio and principal point. In [71, 31], there have been attempts to relax these assumptions, but their formulations can only simultaneously optimize three out of eleven variables in the projection matrix. Additionally, the method in [71] requires the prior knowledge of the height of each human. Other limitations of the mentioned works also prohibit their applications in real world. More specifically, in [70, 71, 38], they assume that the leg-crossing period can be accurately detected. The method in [46] assumes that all objects are moving at constant velocity and the noise model of measurements is known. The work[63] assumes that the variation of relative human heights is sufficiently small. Finally, all the previous methods[70, 71, 46, 42, 132, 48, 63, 38, 9, 31, 112] assume that the camera is not distorted, and their only goal is to estimate the camera projection matrix.

2.2.2 Automatic Radial Distortion Correction

Most existing approaches for automatic radial distortion correction exploit MWA. Devernay *et al.*[26] extract edges in a video sequence and optimize the distortion model such that it can best transform curved edges into straight line segments. The works[8, 130] also attempt to recover straight lines observed in the scene for distortion correction of multiple cameras. As far as we know, the proposed method is the first work that addresses radial distortion correction based on video objects.

2.2.3 Estimation of Distribution Algorithm

The *estimation of distribution algorithm* (EDA) is adopted as the optimization scheme in ESTHER, as well as 2EPOCH. EDA is also known as the *probabilistic model-building genetic algorithm* (PMBGA), which is a category that belongs to the class of *evolutionary algorithms* (EAs). It is inspired from the metaphor of biological evolution. The main difference between EDA and other EAs is that the probability model guiding the search for the optimal solution is explicit instead of implicit. EDA has been applied to some research in image processing, such as fitness evaluation in 3D vehicle modeling[53], but never in the field of camera calibration. In this work, the *estimation of multivariate normal algorithm – global* ($\text{EMNA}_{\text{global}}$)[49, sec. 4.4], a type of multivariate EDA, is adopted for the formulations in ESTHER and 2EPOCH. The advantages of EDA over most other metaheuristics have been reviewed in detail in [36], including its capability to adapt the operators to the problem structure, availability of roadmap in problem solution, prior knowledge exploitation and reduced memory storage. Furthermore, since the sampling of population at each generation can be built into parallel processing, the computation can be highly boosted with GPUs.

2.2.4 Visual Odometry

Visual odometry (VO) is the process of estimating the egomotion of an agent (*e.g.*, vehicle, human, and robot) using only the input from a single or multiple cameras attached to it. Application domains include robotics, wearable computing, augmented reality, and automotive. The term is chosen for its similarity to wheel odometry, which incrementally estimates the motion of a vehicle by integrating the number of turns of its wheels over time. Likewise, VO operates by incrementally estimating the pose of the vehicle through examination of the changes that motion induces on the images from its onboard cameras. For VO to work effectively, there should be sufficient illumination in the environment and a static scene with enough texture to allow apparent motion to be extracted. Furthermore, consecutive frames should be captured by ensuring that they have sufficient scene overlap. The difference from

the stereo scheme is that in the monocular VO, both the relative motion and 3D structure must be computed from 2D bearing data. Since the absolute scale is unknown, the distance between the first two camera poses is usually set to one. As a new image arrives, the relative scale and camera pose with respect to the first two frames are determined using either the knowledge of 3D structure or the trifocal tensor.

The first real-time, large-scale VO with a single camera is presented by Nister *et al.*[87]. They use RANSAC for outlier rejection and 3D-to-2D camera pose estimation to compute the new upcoming camera pose. The novelty of their paper is the use of a five-point minimal solver[86] to calculate the motion hypotheses in RANSAC. After that paper, five-point RANSAC became very popular in VO and has been used in several other works[88, 56]. Corke *et al.*[23] provide an approach for monocular VO based on omnidirectional imagery from a catadioptric camera and optical flow. Lhuillier[56] and Mouragnon *et al.*[81] present an approach based on local windowed-bundle adjustment to recover both the motion and the 3D map (this means that bundle adjustment is performed over a window of the last m frames). Again, they use the five-point RANSAC in [86] to remove outliers.

The major challenge of VO is that some objects in the scene may be moving, *e.g.*, the human objects for 3D pose estimation. To improve robustness, the feature points on moving objects need to be detected and removed.

2.3 Human Pose Estimation

2.3.1 2D Human Pose Estimation

Conventional methods usually solve 2D human poses estimation by tree-structured models, *e.g.*, pictorial structures[94] and mixtures of body parts[140, 15]. These models consist of two terms: a unary term to detect the body joints, and a pairwise term to model the pairwise relationships between two body joints. In [140, 15], a pairwise term is designed as the relative locations and distances between pairs of body joints. The symmetry of appearance between limbs is modeled in [97, 119]. The geometric descriptor greatly reduces

the difficulty for the discriminator in learning domain prior knowledge such as relative limbs length and symmetry between limbs. Recently, impressive advances have been achieved by CNNs[121, 126, 84, 20, 10, 139, 21, 138, 141]. Instead of directly regressing coordinates[121], recent state-of-the-art methods use heatmaps, which are generated by 2D Gaussian models centered on the body joint locations, as the target of regression. 2EPOCH uses OpenPose[10] as the backbone architecture for 2D pose estimation.

2.3.2 3D Human Pose Estimation

Significant progress has been achieved for 3D human pose estimation from monocular images due to the availability of large-scale dataset[6] and the powerful CNNs. These methods can be roughly grouped into two categories.

One-stage approaches directly learn the 3D poses from monocular images. The pioneer work[57] proposes a multi-task framework that jointly trains pose regression and body part detectors. To model high-dimensional joint dependencies, Tekin *et al.*[116] further adopt an autoencoder at the end of the network. Instead of directly regressing the coordinates of the joints, Pavlakoset *et al.*[91] propose a voxel representation for each joint as the regression target, and design a coarse-to-fine learning strategy. These methods heavily depend on fully annotated datasets, and cannot benefit from large-scale 2D pose datasets.

Two-stage approaches first estimate 2D poses and then lift 2D poses to 3D poses[144, 12, 6, 131, 80, 120, 72, 145, 85]. These approaches usually generalize better on images in the wild, since the first stage can benefit from the state-of-the-art 2D pose estimators, which can be trained on images in the wild. The second stage usually regresses the 3D locations from the 2D predictions. For example, Martinez *et al.*[72] propose a simple fully connected residual networks to directly regress 3D coordinates from 2D coordinates. Moreno-Noguer[80] learns a pairwise distance matrix, which is invariant to image rotation, translation, and reflections, from 2D to 3D space. To predict 3D poses for images in the wild, a geometric loss is proposed in [145] to allow weakly supervised learning of the depth regression module. Mehta *et al.*[73, 74] adopt transfer learning to generalize to in-the-wild scenes and they build a

real-time 3D pose estimation solution with kinematic skeleton fitting. 2EPOCH is in the category of two-stage 3D pose estimation. Our design can be applied to a moving camera, and the camera motion can be jointly estimated.

Chapter 3

EVOLUTIONARY SELF-CALIBRATION FROM TRACKING OF HUMANS FOR ENHANCING ROBUSTNESS

Our work on camera self-calibration and automatic radial distortion correction from tracking of walking persons is presented in this chapter, which covers parts of our publications[110, 112, 65, 55, 54, 125, 111]. The proposed framework mainly depends on the evolutionary algorithm to search for the optimal camera parameters and distortion coefficients. The overview of our architecture is shown in figure 3.1.

The proposed method assumes that there is a major planar surface that people can walk on, *i.e.*, the ground plane, in the *field of view* (FOV) of a single static camera. We also require at least one walking human with three different positions, which are not on the same straight line, observable in the scene. An approximate range of the camera height above the ground plane is assumed known. Compared with the assumptions made in other works, our scenario is more realistic.

As shown in figure 3.2, the camera geometry, *i.e.*, WCS, used in this paper is a Cartesian coordinate system in 3D space. The ground plane coincides with the plane defined by the X- and Y-axes. The Z-axis is pointing upward with respect to the ground plane and it passes through the camera position. The camera height is denoted as t_Z.

3.1 Multi-kernel Adaptive Segmentation and Tracking (MAST)

To estimate V_∞ and L_∞ for camera calibration, the first step is to model each human instance as a pole perpendicular to the ground plane, which is equivalent to the localization of head and foot points of each segmented human body. Most previous methods[71, 46, 42, 132, 48, 63, 38, 9] assume they have accurate human tracking and segmentation data as input.

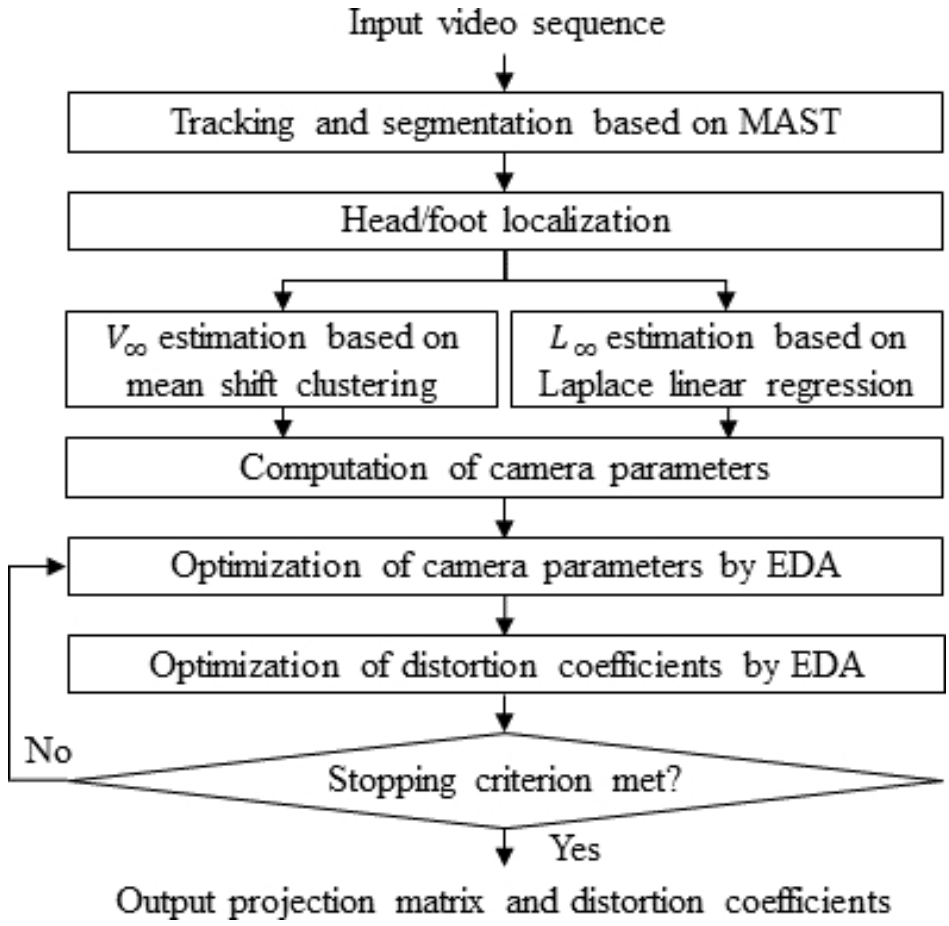

Figure 3.1: This is the flow diagram of ESTHER for camera self-calibration and automatic distortion correction from walking humans.

For more practical usage, instead, we combine the state-of-the-art multi-target tracking and segmentation to support head/foot localization.

The main goal of MAST is to address the problem of *object merging* during tracking by segmentation, *i.e.*, failure in segmentation when some parts of the object(s) share similar color with the background. Figure 3.3 shows the overview flow diagram of the MAST architecture. Each module of the flow chart will be elaborated next.

To begin with, the state-of-the-art change detection scheme, SuBSENSE[106], is adopted

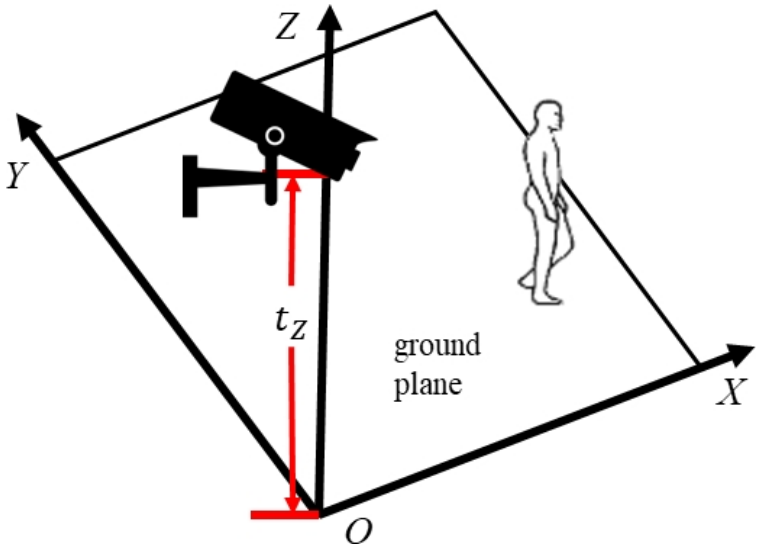

Figure 3.2: The camera geometry is a Cartesian coordinate system in 3D space.

for preliminary object segmentation. Each pixel in the input frame is represented by color (here we choose to use YCbCr space instead of RGB space, since it will facilitate the process of shadow detection) and Local Binary Similarity Patterns (LBSP) feature[104]. The background model is constructed by a set of background samples $B_n(u, v)$ at each pixel location (u, v), which is updated according to an automatically adjusted learning rate. When each new pixel arrives for background/foreground classification, it will be compared with all background samples at the corresponding location. The comparisons are based on two distance thresholds, R_{YCbCr} and R_{LBSP}, in the color space and feature space, respectively. If the number of matched samples (with sufficiently short distance to the input pixel) is smaller than a specific minimum, the pixel is labeled as foreground. To further enhance robustness of SuBSENSE, we add a shadow detection block based on YCbCr color space that starts to function if a pixel is classified as foreground,

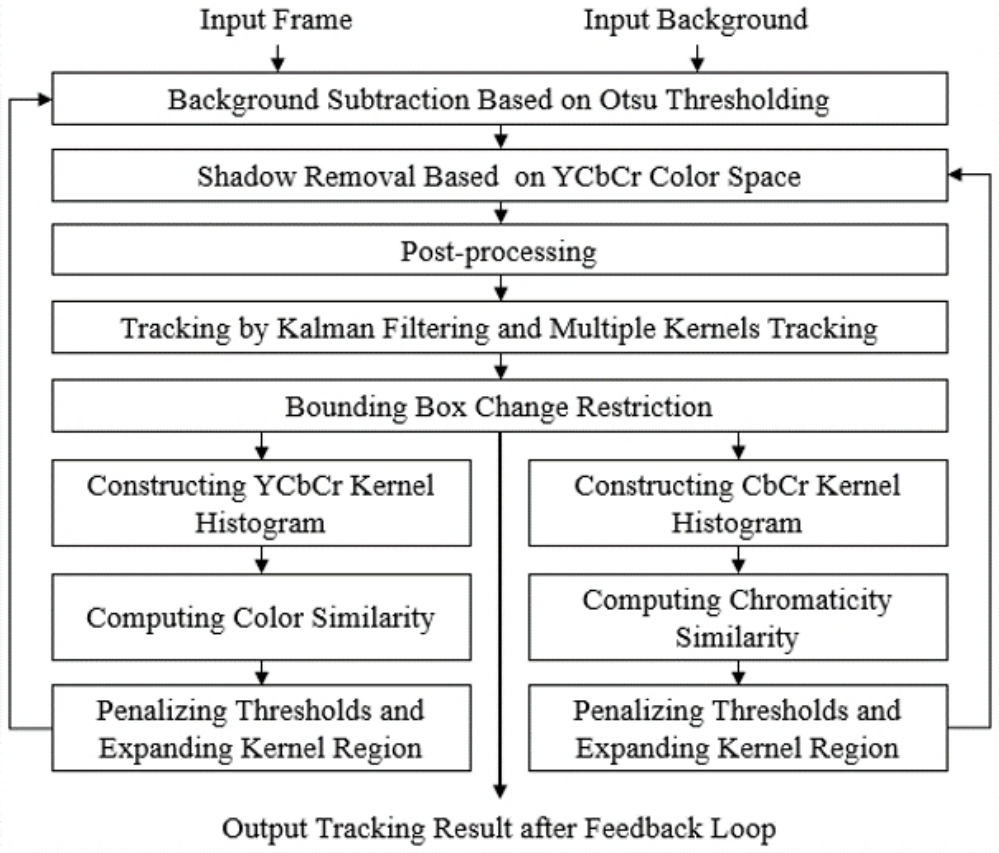

Figure 3.3: This is the flow diagram of MAST for robust object tracking and segmentation.

$$Q_t(u,v) = \begin{cases} 1, & \#\{(\alpha_{\mathrm{Y}} \le I_t^{\mathrm{Y}}(u,v)/B_n^{\mathrm{Y}}(u,v) \le \beta_{\mathrm{Y}}) \\ & \bigwedge(|I_t^{\mathrm{Cb}}(u,v) - B_n^{\mathrm{Cb}}(u,v)| \le \tau_{\mathrm{Cb}}) \\ & \bigwedge(|I_t^{\mathrm{Cr}}(u,v) - B_n^{\mathrm{Cr}}(u,v)| \le \tau_{\mathrm{Cr}}), \forall n\} > N_{\max} \\ 0, & \text{otherwise} \end{cases}, \tag{3.1}$$

where $Q_t(u,v)$ indicates shadow when the value is 1, $I_t(u,v)$ is a pixel from current frame t, the superscripts of $I_t(u,v)$ and $B_n(u,v)$ indicate the YCbCr channels, $N_{\max}$ is the maximum number of matches required for shadow detection, and α_{Y}, β_{Y}, τ_{Cb} and τ_{Cr} are the thresholds

for their corresponding color channels. If a pixel is detected as shadow, it is discarded from foreground, and will be used for updating the background model. After segmentation, morphological operations, *e.g.*, closing, opening, and flood-filling, are further applied on the derived foreground mask for shape refinement.

In the segmented foreground, each object blob may contain more than one target, *i.e.*, the problem of *initial occlusion*. Thus, an HOG human detector[24], which has been selected empirically because of its efficiency and sufficient accuracy, is run on the cropped frame image within each object bounding box. If multiple targets are detected and their overlapping area with each other is small enough, they will be initialized separately for object tracking. Different from conventional tracking by detection that needs to process each entire frame image, the computation complexity is much reduced since only the local region around each foreground blob is considered.

Based on the observations from segmentation and local object detection, we can start tracking each target. The preliminary tracking results are generated by the method proposed by Chu *et al.*[18] that combines Kalman filtering and CMK tracking. Kalman filter prediction is first conducted on all the objects tracked in the previous frame. Then we detect whether there is abnormality in size change of each foreground blob, which can be caused by occlusion or failure in segmentation. The abnormal targets and those initialized by object detection are tracked by the CMK method, which relies on multiple inter-related kernels to represent different parts of human, so that we can add weights of trust on different kernels depending on their severity of occlusion. Multiple measurements are produced from CMK tracking that are handled by probabilistic data association. On the other hand, each normal foreground blob with single object is directly selected as the measurement for Kalman filtering.

From preliminary tracking results, we make use of multiple kernels to measure similarity between current frame and background in the object regions. In our experiments, each human target is described by two kernels that cover half of his/her body on the top and bottom respectively, as people usually wear differently in these two body parts. Two kernel histograms are constructed within each kernel region for both current frame and background

model: one of them is built in the YCbCr color space, and the other only uses the Cb and Cr channels to represent the chromaticity information. Note that the kernel histograms for background use all background samples and are normalized for comparison. To emphasize the object region that usually covers the central area of each kernel, the kernel histograms are weighted by a Gaussian function,

$$w_{\text{ker}} = \frac{1}{2\pi\sigma_u\sigma_v}\exp[-\frac{u-u_m}{2\sigma_u^2} - \frac{v-v_m}{2\sigma_v^2}], \tag{3.2}$$

in which σ_u and σ_v are set as half of the width and height of the kernel bounding box, respectively, whereas u_m and v_m locate the mean point of the foreground shape within the kernel.

Afterwards, the color similarity and chromaticity similarity are respectively computed as the reciprocals of Bhattacharyya distances between corresponding kernel histograms, h_{YCbCr} and h_{CbCr},

$$simi_{\text{color}} = 1/\sum_c \sqrt{h^I_{\text{YCbCr}}(c) \cdot h^B_{\text{YCbCr}}(c)}, \tag{3.3}$$

$$simi_{\text{chrom}} = 1/\sum_c \sqrt{h^I_{\text{CbCr}}(c) \cdot h^B_{\text{CbCr}}(c)}, \tag{3.4}$$

where superscripts I and B denote the kernel histograms in current frame and background, respectively, and c is the index of channel bin. The higher the color similarity of object region with background, the more likely the object will mistakenly merge into background during segmentation. Likewise, if the object region shares high similarity in chromaticity with the background, *e.g.*, a human wearing black pants is walking on a grey ground plane, it is easy for his/her body parts to be wrongly recognized as shadow and removed from foreground.

Next, a second segmentation using thresholding parameters penalized by $simi_{\text{color}}$ and $simi_{\text{chrom}}$ is conducted in order to preserve more foreground in the local region around tracked targets. Under the consideration of smoothness, the penalty weights on segmentation thresholds are computed based on a fuzzy Gaussian penalty weighting function as follows,

$$w_{\text{pen}} = \begin{cases} \exp[-\frac{9\cdot(1.0-simi)^2}{4\cdot(1.0-simi_{\text{min}})^2}], & simi_{\text{min}} \leq simi_{\text{max}} \\ 0, & \text{otherwise} \end{cases}, \tag{3.5}$$

in which $simi$ is the color or chromaticity similarity computed from equation 3.3 or equation 3.4 respectively, whereas $simi_{\text{min}}$ and $simi_{\text{max}}$ represent the range of $simi$ value for re-segmentation. As shown in the fuzzy Gaussian curve in figure 3.4, when $simi$ is smaller than the lower bound $simi_{\text{min}}$, the preliminary segmentation is considered successful, and there is no need for further adaptation This is based on the concept of fuzzy set. On the contrary, if $simi$ is too large, it is highly likely to be caused by tracking error, where CMK tracking wrongly shifts to a background area. Hence, to prevent propagation of errors, an upper bound $simi_{\text{max}}$ is defined for the similarity between current frame and background. The w_{pen} computed based on $simi_{\text{color}}$ is used to penalize R_{YCbCr} and R_{LBSP} in SuBSENSE, whereas the one for $simi_{\text{chrom}}$ is applied on τ_{Cb} and τ_{Cr} in shadow detection. The penalization is defined by multiplying $(1 - w_{\text{pen}})$. Besides, since the preliminary foreground blob may fail to cover the entire object body, the kernel region to conduct re-segmentation is expanded by a factor of $w_{\text{pen}}/2$. In summary, the adaptive segmentation is operated in a larger kernel region with lower thresholds for background subtraction and shadow detection. Therefore, the segmented foreground area is expanded to maintain continuity of tracking by segmentation. The final foreground mask is created by a union combination of the first segmentation across the entire frame and the local adaptive segmentation in selected kernel regions.

Lastly, the tracking module is executed again to generate the final tracking results from the updated foreground mask. Note that Kalman filter update is not performed until after re-segmentation. The superiority of adding multi-kernel feedback loops to adaptively control the segmentation parameters can be seen from figure 3.5, in which more foreground belonging to the target is retained by MAST, even when the chromaticity of her clothing is similar to the background.

The procedure of head/foot localization is demonstrated in figure 3.6. From the output of MAST, the bounding box and segmented foreground blob for each object instance can be

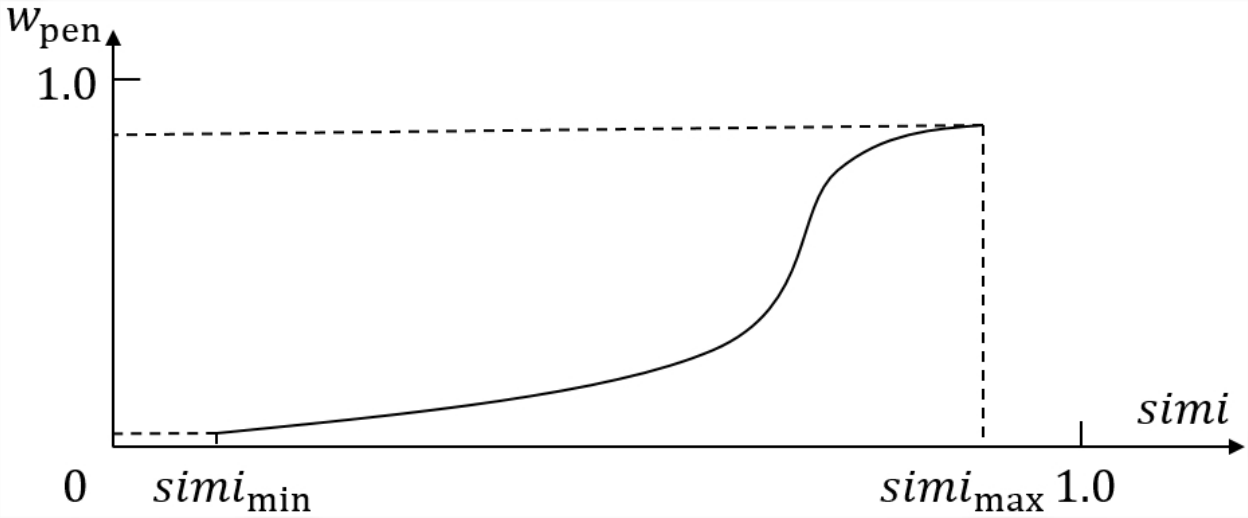

Figure 3.4: This is the fuzzy Gaussian function for computing the penalty weight against the similarity between the current frame and the background.

derived. We compute the first moment of each foreground blob to determine its major axis. Each human instance can therefore be approximated as a pole representing its orientation. The two intersecting points between the major axis and the bounding box are chosen as the head and foot locations. This scheme has also been effectively applied in several other works[63, 31, 14].

3.2 Estimation of Vanishing Points

All the instances of humans can be modeled as poles perpendicular to the ground plane. Ideally, if all the head and foot points are located correctly, *i.e.*, there is neither noise nor outlier, and there is no radial distortion, V_∞ and L_∞ can be easily determined as illustrated in figure 3.7. The straight lines passing through the head and foot points at all object instances should converge at one point, *i.e.*, the vertical vanishing point, V_∞. Similarly, if we draw a straight line to connect the head points of the same object at two different instances and another straight line connecting their foot points, the intersection of the two lines should lie

Figure 3.5: This is the comparison of segmentation performance. On the left shows the segmentation from the preliminary results of SuBSENSE with shadow detection. On the right shows the segmentation after the application of multi-kernel feedback loops. Note that foreground is colored in red, and detected shadow in blue.

on the horizon line, L_∞, which is defined as the extension of the ground plane at infinity. However, due to the existence of noise and outliers, this scenario is unrealistic in real world. There are always many candidate points of V_∞, each generated by a pair of object instances. Similarly, the candidate points of L_∞ may not lie on the same straight line.

To estimate the location of V_∞, we propose a method based on mean shift clustering. The sensitivity to noise in head/foot localization is usually high for V_∞ estimation, because each object instance is associated with all the others in the point set of V_∞ candidates. Since the number of outliers can easily overwhelm inliers in most cases, the performance of RANSAC is not sufficiently robust. The problem can be better solved by applying mean shift clustering, because when spatially close clusters are merged together, the shape of the final cluster of inliers is not constrained. On the contrary, the cluster of inliers in RANSAC must form a circle. More specifically, the estimation of V_∞ is defined as

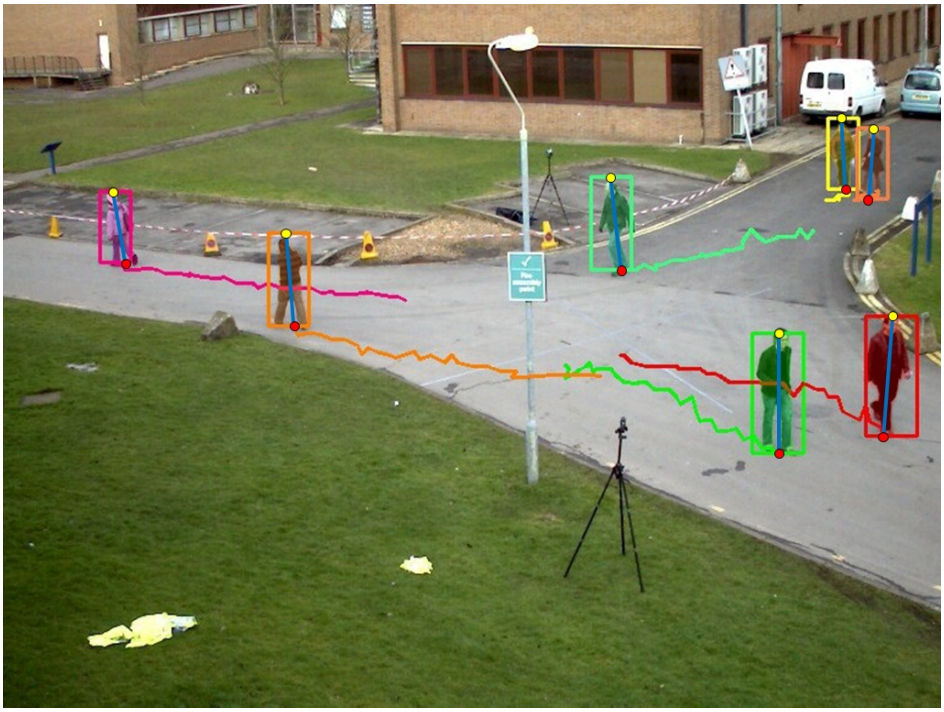

Figure 3.6: This is the demonstration of head/foot localization from tracking and segmentation. The colored rectangles are bounding boxes from 2D tracking. The segmented foreground masks are also overlaid in color. The blue line segments denote the major axes of the foreground blobs. The yellow and red dots respectively denote the located head and foot points.

$$\begin{aligned} V_\infty &= \text{mean}(C^*), \\ \text{s.t.,} \quad C^* &= \arg\max_{C \in \{C\}} \#(C), \end{aligned} \tag{3.6}$$

where C denotes each cluster in mean shift clustering. The functions mean$(\cdot)$ and $\#(\cdot)$ respectively represent the computations of mean point and the number of candidate points. The mean shift window bandwidth is empirically set as $BW = 1 \times 10^3$ pixels in our experiments. In every iteration, an unvisited V_∞ candidate is randomly selected as the initial mean

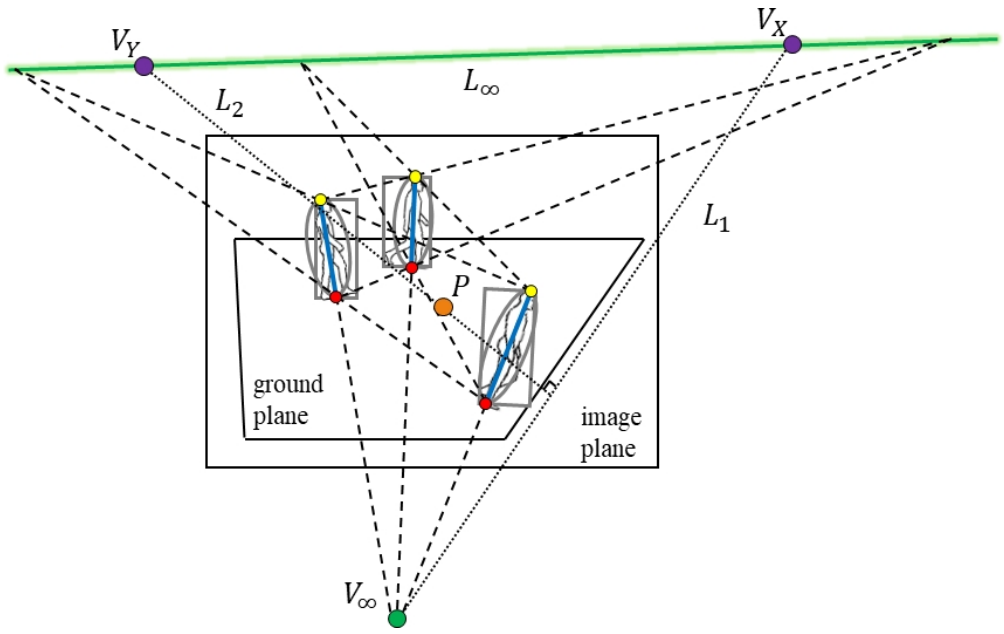

Figure 3.7: This is the geometry of vanishing points estimation in self-calibration (ideal scenario). The three blue poles represent three instances of the same person appearing at different frames, with yellow and red dots indicating their head and foot points, respectively. Point V_∞ in green marks the vertical vanishing point. The green line L_∞ denotes the horizon line. The dashed lines are auxiliary lines for the computation of V_∞ and L_∞. Points V_X and V_Y in purple are the other two vanishing points located on L_∞. Point P in orange gives the principal point of the camera. The dotted lines L_1 and L_2 are auxiliary lines to locate V_Y.

point. Then we conduct mean shift based on the window bandwidth BW until the moving distance of each mean point is smaller than a threshold $\tau_{BW} = BW \times e^{-3}$. The iteration is repeated until there is no more unvisited point left. Then, all the clusters whose mean points are within $BW/2$ are merged together. The estimated V_∞ is chosen as the mean point of the cluster with the most inliers, whereas other clusters are treated as outliers.

Laplace linear regression is proposed for the estimation of L_∞. Another drawback of the traditional method based on RANSAC is that the threshold parameter needs to be fine-tuned

depending on different camera views. Hence, we leverage robust linear regression based on probabilistic modeling to avoid this configuration. The noise modeling by linear regression using Gaussian distribution can perform poorly when there are outliers in the data. As deviations are penalized quadratically by squared error, outliers will have greater influence on the line fitting than inliers. On the other hand, if we use Laplace distribution, its heavy tails can enforce higher likelihood to be assigned to points far away without the need to perturb the line[82, sec. 7.4]. Therefore, the results will be more robust. The likelihood model of Laplace linear regression is given as

$$p(\mathbf{v}|\mathbf{u},\mathbf{w}) = \text{Laplace}(\mathbf{v}|\mathbf{w}^T\mathbf{u}) \propto \exp(-|\mathbf{v}-\mathbf{w}^T\mathbf{u}|), \tag{3.7}$$

where $\mathbf{u}$ and $\mathbf{v}$ are the vectors containing the 2D coordinates of the candidate points for L_∞, and $\mathbf{w}$ represents the parameters of L_∞ that we aim to estimate. This problem can be formulated as constrained optimization,

$$\begin{aligned} \min_{\mathbf{w},\mathbf{r}} \sum_l r_l &= \min_{\mathbf{w},\mathbf{r}^+,\mathbf{r}^-} \sum_l (r_l^+ + r_l^-), \\ \text{s.t.,} \quad r_l^+ \geq 0, \quad r_l^- \geq 0, \quad & \mathbf{w}^T u_l + r_l^+ - r_l^- = v_l, \end{aligned} \tag{3.8}$$

in which $r_l \triangleq r_l^+ - r_l^-$ is the lth residual that can be split into positive and negative residuals, so that the objective function becomes a linear objective. This problem can be solved by linear programming solvers such as CVX[33]. The standard formulation is as follows,

$$\min_{\boldsymbol{\theta}} \mathbf{f}^T\boldsymbol{\theta}, \quad \text{s.t.,} \quad \mathbf{A}\boldsymbol{\theta} \leq \mathbf{b}, \quad \mathbf{A}_{\text{eq}}\boldsymbol{\theta} \leq \mathbf{b}_{\text{eq}}, \quad \mathbf{LB} \leq \boldsymbol{\theta} \leq \mathbf{UB}, \tag{3.9}$$

where $\boldsymbol{\theta} = (\mathbf{w}, \mathbf{r}^+, \mathbf{r}^-)$, $\mathbf{f} = [\mathbf{0}, \mathbf{1}, \mathbf{1}]$, $\mathbf{A} = []$, $\mathbf{b} = []$, $\mathbf{A}_{\text{eq}} = [\mathbf{u}, \mathbf{I}, -\mathbf{I}]$, $\mathbf{b}_{\text{eq}} = \mathbf{v}$, $\mathbf{LB} = [-\infty\mathbf{1}, \mathbf{0}, \mathbf{0}]$, and $\mathbf{UB} = []$.

Finally, based on the estimated V_∞ and L_∞, the other two vanishing points that lie on L_∞, namely V_X and V_Y, can be computed. As demonstrated in figure 3.7, first we initialize the location of the principal point P at the center of the image. The optimization for a more accurate location of P will be addressed later in this chapter. The next step is to randomly

locate a V_X on L_∞. Then we draw an auxiliary line L_1 that connects V_X and V_∞, and another line L_2 that is perpendicular to L_1 and passes through P. Since the principal point of a camera should be the orthocenter of the triangle formed by three vanishing points[11], V_Y can be located at the intersection between L_∞ and L_2.

3.3 Computation of Camera Parameters

In a general pinhole camera model, the goal of camera calibration is to find a 3×4 projection matrix $\mathbf{P}$ that can project every 3D point (X, Y, Z) to its corresponding 2D pixel location (u, v) by

$$[u, v, 1]^T \sim \mathbf{P} \cdot [X, Y, Z, 1]^T. \tag{3.10}$$

This projection matrix can be decomposed into three matrices, including the intrinsic parameter matrix $\mathbf{K}$ that contains five intrinsic parameters (focal length in u-direction f_u, focal length in v-direction f_v, coordinates of principal point c_u and c_v, and skew s), the rotation matrix $\mathbf{R}$ defined by three extrinsic parameters (roll angle around Z-axis γ, pan angle around Y-axis α, and tilt angle around X-axis β), as well as the translation matrix $\mathbf{t}$ with the other three extrinsic parameters (translation along X-axis t_X, translation along Y-axis t_Y, and translation along Z-axis t_Z). Their relations are provided below:

$$\mathbf{P} = \mathbf{K} \times [\mathbf{R}|\mathbf{t}],$$
$$\text{s.t.,} \quad \mathbf{K} = \begin{bmatrix} f_u & s & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} t_X \\ t_Y \\ t_Z \end{bmatrix} \quad \text{and} \quad \mathbf{R} = \mathbf{R}_Z \cdot \mathbf{R}_Y \cdot \mathbf{R}_X,$$
$$\mathbf{R}_Z = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{R}_Y = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix}, \tag{3.11}$$
$$\mathbf{R}_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\beta & -\sin\beta \\ 0 & \sin\beta & \cos\beta \end{bmatrix}.$$

Based on the assumptions on fixed intrinsic camera parameters[71, 46, 42, 132, 48, 63, 38, 9, 31], *i.e.*, $f_u = f_v$, (c_u, c_v) located at the image center and $s = 0$, the camera parameters can be computed from given locations of P, V_X and V_Y as follows.

$$\gamma = \tan^{-1}\left(\frac{v_{V_Y} - v_{V_X}}{u_{V_X} - u_{V_Y}}\right), \tag{3.12}$$

$$\begin{aligned} f_u = f_v &= \sqrt{-(v^{\text{rot}}_{V_X} \cdot v^{\text{rot}}_{V_Y} + u^{\text{rot}}_{V_X} \cdot u^{\text{rot}}_{V_Y})}, \\ \text{s.t.,} \quad v^{\text{rot}}_{V_X} &= \cos\gamma(v_P - v_{V_X}) - \sin\gamma(u_{V_X} - u_P), \\ v^{\text{rot}}_{V_Y} &= \cos\gamma(v_P - v_{V_Y}) - \sin\gamma(u_{V_Y} - u_P), \\ u^{\text{rot}}_{V_X} &= \cos\gamma(u_{V_X} - u_P) + \sin\gamma(v_P - v_{V_X}), \\ u^{\text{rot}}_{V_Y} &= \cos\gamma(u_{V_Y} - u_P) + \sin\gamma(v_P - v_{V_Y}), \end{aligned} \tag{3.13}$$

$$\beta = -\tan^{-1}\left(\frac{v^{\text{rot}}_{V_X}}{f_u}\right), \tag{3.14}$$

$$\alpha = -\tan^{-1}\left(\frac{\cos\beta \cdot u^{\text{rot}}_{V_X}}{f_u}\right). \tag{3.15}$$

According to the camera geometry in figure 3.2, the translation parameters t_X and t_Y are zero. And t_Z is equal to the camera height, whose approximate range is assumed known. The camera parameters will be further optimized by EDA.

3.4 Optimization of Camera Parameters by EDA

As discussed in the review paper[79], the major limitation of all self-calibration methods based on the estimation of V_∞ and L_∞ is their unrealistic assumptions on unknown intrinsic camera parameters, which give rise to increasing reprojection error. To relax these assumptions, we formulate the optimization of camera parameters based on the minimization of reprojection error on the ground plane.

To start with, a set of $n_X \times n_Y$ grid points are generated on the ground plane in 3D space, *i.e.*, the XY-plane. The rows and columns are parallel with X- and Y-axes respectively. The 3D coordinates of the grid points are denoted as $(X_i, Y_j, 0)$, where $i = 0, 1, \ldots, n_{X-1}$ and $j = 0, 1, \ldots, n_{Y-1}$. Using the initial camera parameters computed by estimated V_X and V_Y, the grid points can be projected to 2D (see figure 3.8). Their corresponding projected 2D pixel locations are $p_{i,j}$. According to the definition of vanishing points, if we generate n_Y straight lines, noted l_j^X, on the image plane that each connects V_X with one of the points $p_{0,j}$ on the edge of the grid, all the other grid points $p_{i,}(i > 0, j > 0)$ should fall on these lines. However, due to reprojection error, some $p_{i,j}$ may not lie on l_j^X, and the Euclidean distance between them is denoted as $d_{i,j}^X$. Similarly, the distance between $p_{i,j}$ and the corresponding straight line l_i^Y that connects V_Y and $p_{i,0}$ is denoted as $d_{i,j}^Y$. Now we can define the objective function of this optimization problem by

$$\begin{aligned} \mathbf{P}^* &= \underset{\mathbf{P}\in \mathrm{Rng}_{\mathbf{P}}}{\arg\min}\, \mathrm{E}\left[d_{i,j}^X + d_{i,j}^Y\right], \\ \text{s.t.,} \quad d_{i,j}^X &= \|l_j^X, p_{i,j}\|_2, \quad d_{i,j}^Y = \|l_i^Y, p_{i,j}\|_2, \end{aligned} \tag{3.16}$$

where $\mathrm{E}(\cdot)$ computes the expected value that is equivalent to the reprojection error on the ground plane. The function $\|\cdot\|_2$ measures Euclidean distance in pixel. In our formulation, $\mathbf{P}$ is decomposed into 11 camera parameters to be optimized. The initial range for camera

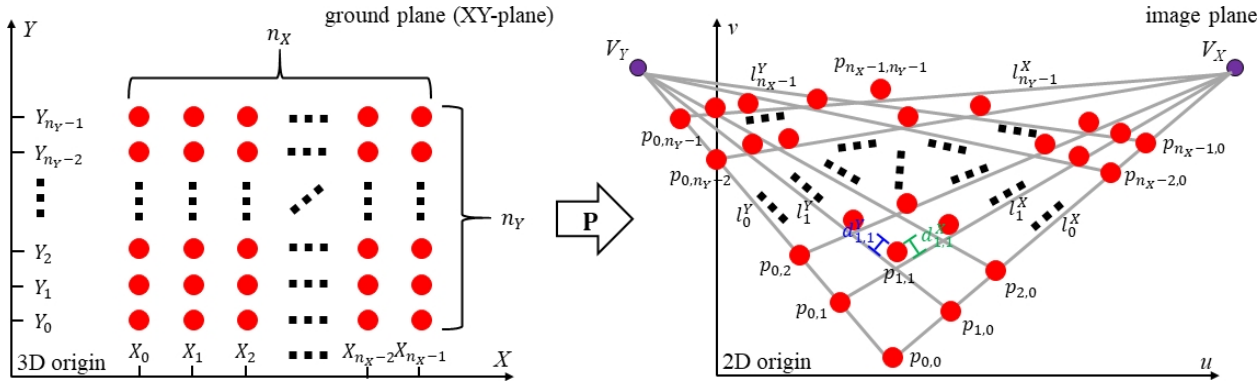

Figure 3.8: This is the illustration of the optimization of camera parameters based on EDA. A set of $n_X \times n_Y$ grid points on the 3D ground plane are projected to 2D. Because of reprojection error, a projected 2D point $p_{i,j}(i > 0, j > 0)$ may not locate at the intersection between l_j^X, connecting V_X with $p_{0,j}$, and l_i^Y, connecting V_Y with $p_{i,0}$. Hence, we can use the distances from $p_{i,j}$ to these two lines, $d_{i,j}^X$ and $d_{i,j}^Y$, to indicate the reprojection error that we aim to minimize.

parameters, noted $\text{Rng}_{\mathbf{P}}$, is empirically set as $0.02 \times f_u$ for f_u, $0.02 \times f_v$ for f_v, 20 pixels for c_u and c_v, 20 degrees for γ, β and α, 200 mm for t_Z, and 0 for s, t_X and t_Y.

The optimization problem is formulated as multivariate EDA. We search for the local optima of camera parameters, which induce minimal reprojection error, simultaneously in $\text{Rng}_{\mathbf{P}}$. Therefore, the assumptions on intrinsic camera parameters can be relaxed effectively. We do not need to know the real human heights and other measurements in the scene for this formulation. Following the conventional EDA pseudocode[49], the detailed algorithmic procedure is illustrated in algorithm 1. The sizes of initial and selected populations are empirically set as 2,000 and 20, respectively. The stopping criterion is that the decreasing ratio of the reprojection error is smaller than a threshold τ_r or the number of generations is larger than $g_{\max}$. In our experiments, we set $\tau_r = 0.1$ and $g_{\max} = 100$. As for the 3D grid on the ground plane, its size is empirically set as 10×10, where each grid point is 1 meter

away from its neighbors.

Algorithm 1 Optimization of camera parameters by EDA

Require: initial range $\text{Rng}_{\mathbf{P}}$, size of initial population R, size of selected population $N < R$, maximum number of generations $g_{\max}$, stopping threshold of decreasing ratio τ_r

Ensure: optima of camera parameters $\mathbf{P}^*$

1: generate initial population $\mathcal{P}(0) \leftarrow R$ sets of parameters sampled uniformly in the 11D space within $\text{Rng}_{\mathbf{P}}$;
2: $g \leftarrow 0$;
3: **while** ($g > 1$ and $\frac{\mu_{g-2} - \mu_{g-1}}{\mu_{g-2}} > \tau_r$) and $g < g_{\max}$ **do**
4: acquire each set of parameters from $\mathcal{P}(g)$;
5: project $n_X \times n_Y$ 3D grid on the ground plane to 2D;
6: measure error distance $d_{i,j}^X$ from each $p_{i,j}$ to l_j^X;
7: measure error distance $d_{i,j}^Y$ from each $p_{i,j}$ to l_j^Y;
8: select the population of promising solutions $\mathcal{S}(g) \leftarrow N$ individuals within $\mathcal{P}(g)$ that have smaller cost values $c_g = \text{E}\left[d_{i,j}^X + d_{i,j}^Y\right]$;
9: build probabilistic model $\mathcal{M}(g) = \mathcal{N}(\mu_g, \sigma_g) \leftarrow$ eleven-variate normal density function modeled from $\mathcal{S}(g)$;
10: $\mathcal{P}(g+1) \leftarrow R$ individuals sampled from $\mathcal{M}(g)$;
11: $g \leftarrow g + 1$;
12: **end while**
13: output μ_g of $\mathcal{M}(g)$.

3.5 Radial Distortion Correction by EDA

A major problem of the above procedure is that it does not work for a camera suffering from radial distortion, where L_∞ becomes a nonlinear curve and the vertical poles would not converge at V_∞. To address this issue, we propose to optimize the distortion coefficients by EDA that will enable camera self-calibration for wide-angle cameras.

For a pixel point (u, v) in a distorted frame image, the corrected pixel point (u', v') can be represented as

$$\begin{aligned} u' =& u(1 + k_1 r^2 + k_2 r^4 + k_3 r^6), \\ v' =& v(1 + k_1 r^2 + k_2 r^4 + k_3 r^6), \\ & \text{s.t.}, r^2 = u^2 + v^2, \end{aligned} \tag{3.17}$$

where $\mathbf{k} = [k_1, k_2, k_3]^T$ is the vector of distortion coefficients to be estimated.

The projection of head and foot points to their corresponding pixel locations is given as

$$\begin{aligned} [\lambda u'_{\text{head}}, \lambda v'_{\text{head}}, \lambda]^T &= \mathbf{P} \cdot [X_{\text{foot}}, Y_{\text{foot}}, H, 1]^T, \\ [\lambda u'_{\text{foot}}, \lambda v'_{\text{foot}}, \lambda]^T &= \mathbf{P} \cdot [X_{\text{foot}}, Y_{\text{foot}}, 0, 1]^T, \end{aligned} \tag{3.18}$$

where λ is the scale factor and H is the human height in 3D space. The X- and Y-coordinates of head and foot points are considered the same, because we assume that a human body is always standing upright on the ground plane.

As illustrated in figure 3.9, it is intuitive that the estimated 3D height of the same walking person can vary largely when the camera is under radial distortion. Thus, the objective function of this optimization problem is designed as,

$$\begin{aligned} \mathbf{k}^* &= \underset{\mathbf{k} \in \text{Rng}_{\mathbf{k}}}{\arg\min}\, \text{E}\left[\Delta H_{o,t}^2\right], \\ \text{s.t.,} \quad \Delta H_{o,t} &= \frac{\Delta H_{o,t} - \overline{H_o}}{\overline{H_o}}, \end{aligned} \tag{3.19}$$

where $\text{E}(\cdot)$ computes the expected value and $\Delta H_{o,t}$ is the relative human height offset of the o'th object at the t'th frame. The human height $H_{o,t}$ can be solved from equation 3.18. The mean of the o'th object's estimated heights is denoted as $\overline{H_o}$. $\text{Rng}_{\mathbf{k}}$ is the initial range for the optimization of $\mathbf{k}$. The normalization for the relative human height is to mitigate the influence of height difference of different people. This nonlinear optimization problem can be solved using EDA, where the probabilistic model is a three-variate normal distribution.

The detailed algorithmic procedure is described in the form of pseudocode in algorithm 2.

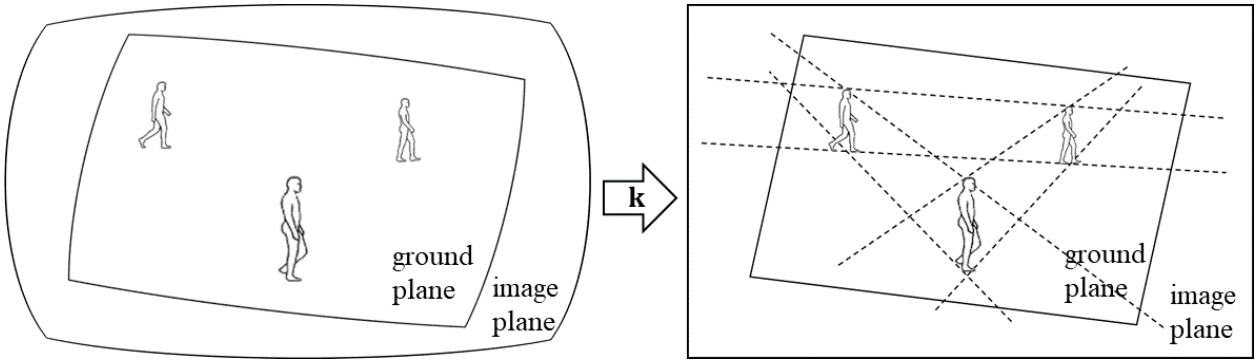

Figure 3.9: This illustrates radial distortion correction by EDA optimization. The vector of radial distortion coefficients, $\mathbf{k}$, is optimized based on the minimization of human height variance.

The distortion coefficients will gradually converge to the values that generate the lowest variance of the relative human heights. All the configuration settings are the same as algorithm 1, except the initial range $\text{Rng}_{\mathbf{k}}$, which is set as 0.5 for k_1, 5.0 for k_2 and 0 for k_3.

Algorithm 1 and algorithm 2 can be considered as a two-step evolutionary optimization process. They are repeated iteratively until the stopping criterion is met, *i.e.*, the decreasing ratio of relative human height variance is smaller than a specific threshold, which is empirically set as 0.01.

3.6 Experimental Results

To evaluate the performance of the proposed method, we conduct various experiments on video sequences from three public benchmarks and our own captured dataset. There are two video sequences, *Outdoor* and *Indoor*, from the VPTZ benchmark[95] for virtual PTZ camera simulation and a sequence, *Terrace*, from the EPFL benchmark[29] for multi-camera pedestrian detection and tracking. These three sequences have also been adopted in the work[9] for experimental comparison. Besides, we use two video sequences from the *MOTChallenge* 3D benchmark[50], *PETS09-S2L1* and *AVG-TownCentre*. They have been used in[31] for

Algorithm 2 Radial distortion correction by EDA

Require: initial range $\text{Rng}_{\mathbf{k}}$, size of initial population R, size of selected population $N < R$, maximum number of generations $g_{\max}$, stopping threshold of decreasing ratio τ_r, optimized projection matrix $\mathbf{P}^*$

Ensure: optimal distortion coefficients $\mathbf{k}^*$

generate initial population $\mathcal{P}(0) \leftarrow R$ sets of distortion coefficients sampled uniformly in the 3D space within $\text{Rng}_{\mathbf{k}}$;
$g \leftarrow 0$;
while ($g > 1$ and $\frac{\mu_{g-2} - \mu_{g-1}}{\mu_{g-2}} > \tau_r$) and $g < g_{\max}$ **do**
    acquire each set of coefficients from $\mathcal{P}(g)$;
    correct pixel locations by equation 3.17;
    estimate the 3D human height of each instance by solving equation 3.18 with $\mathbf{P}^*$;
    calculate $\Delta H_{o,t}$ in equation 3.19;
    select the population of promising solutions $\mathcal{S}(g) \leftarrow N$ individuals within $\mathcal{P}(g)$ that have smaller cost values $c_g = \text{E}\left[\Delta H_{o,t}^2\right]$;
    build probabilistic model $\mathcal{M}(g) = \mathcal{N}(\mu_g, \sigma_g) \leftarrow$ three-variate normal density function modeled from $\mathcal{S}(g)$;
    $\mathcal{P}(g+1) \leftarrow R$ individuals sampled from $\mathcal{M}(g)$;
    $g \leftarrow g + 1$;
end while
output μ_g of $\mathcal{M}(g)$.

ID	Seq.	Dataset	Res. (pix.)	Hz (fps)	Len. (s)	No. objects
1	*Outdoor*	VPTZ[95]	1280×960	15	400	20
2	*Indoor*	VPTZ[95]	1280×960	15	60	10
3	*Terrace*	EPFL[29]	360×288	25	200	8
4	*PETS09-S2L1*	*MOTChallenge*[50]	768×576	7	114	19
5	*AVG-TownCentre*	*MOTChallenge*[50]	1920×1080	2.5	225	226
6	*Soccer-S1*	Ours	2048×1536	25	120	16
7	*Soccer-S2*	Ours	2048×1536	25	120	16
8	*Soccer-S3*	Ours	1280×720	25	120	16
9	*Soccer-S4*	Ours	2048×1536	25	120	16

Table 3.1: This table lists the details of experimental video sequences for camera self-calibration and radial distortion correction.

the evaluation of self-calibration and 3D tracking. Finally, to emphasize our effectiveness in radial distortion correction, we also include four sequences that are synchronously recorded at a soccer game by fish-eye cameras. They are respectively denoted as *Soccer-S1*, *Soccer-S2*, *Soccer-S3* and *Soccer-S4*. The details of all test sequences are summarized in table 3.1.

3.6.1 Comparison of Camera Self-calibration

The proposed method, ESTHER, is compared with several state-of-the-art approaches in camera self-calibration from human tracking listed as follows:

- Our earlier method[112], which does not include radial distortion correction based on the minimization of human height variance;

- The method by Führ *et al.*[31] which employs RANSAC for noise reduction and formulates a nonlinear optimization problem based on the reprojected poles of walking humans;

- A recent method by Brouwers *et al.*[9], which adds a pre-processing step to filter away detection outliers and a post-processing step for tilt angle optimization based on human height distribution;

- The method by Liu *et al.*[63] that utilizes predicted human height distribution to optimize the focal length;

- Another method by Liu *et al.*[64] based on [63] but leverages multi-camera information;

- The method by Wu *et al.*[132] that employs RANSAC for noise reduction in the estimation of V_∞ and L_∞;

- The original method by Lv *et al.*[70] without any scheme of noise reduction or optimization.

The works[112, 31, 9, 64] are the state-of-the-art in this field. The experimental results of [31, 9, 63, 64] are derived from their published papers. As for [112, 132, 70] and the proposed method, the head and foot points located from MAST are used as their input, where the default configuration parameters for MAST and SuBSENSE are applied. For [132], the RANSAC threshold for L_∞ estimation is fine-tuned for each video sequence, and the corresponding threshold for V_∞ estimation is set to be the same as our bandwidth in mean shift clustering, *i.e.*, 1×10^3 pixels.

The experimental results of camera calibration on each of the nine test sequences are presented in table 3.2, table 3.3 and table 3.4. For evaluation, we measure the absolute differences between the ground truths and estimated camera parameters, including f (the average of f_u and f_v), c_u, c_v, γ, β, and t_Z. The proposed method, ESTHER, shows the best overall performance across all the metrics. The qualitative performance of ESTHER is displayed in figure 3.10. Especially, we demonstrate significant improvement on videos with strong radial distortion, *i.e.*, Seq. #1, #2, #5, #6, #7, #8 and #9, which validates the effectiveness of the proposed scheme for radial distortion correction based on evolutionary

optimization. In the other test sequences, *i.e.*, Seq. #3 and #4, because the distortion effect is minor, our previous approach[112] also achieves robust performance. Thus, the advantage of relaxing assumptions on unknown intrinsic parameters is validated. All the other approaches assume that the principal point locates at the center of the frame image, but in most scenarios, there is a non-negligible distance between these two points. In our algorithm, however, the principal point coordinates can be effectively optimized through the minimization of reprojection error. Besides, we also generate better estimation of the focal length by relaxing the constraint on aspect ratio. The performance of the method[31] on Seq. #4 and #5 is comparable to ours, due to the similar nonlinear optimization of camera parameters. The experimental results by Brouwers *et al.*[9] are only available on the first three video sequences. Because of their extra processing steps that fine-tune the rotation angles, they perform better in the estimation of γ and β, but the computation of the other camera parameters is less reliable. As for the method by Liu *et al.*[63, 64], they only compare their performance of focal length estimation on the *Outdoor* sequence. Though the cues from multiple cameras can be leveraged to improve estimation accuracy in [64], the final results are still far from matching our expectation. With noise removal by RANSAC, Wu *et al.*[132] enhance the reliability of the original work[70], but due to the lack of optimization process, their method fails in most cases. Finally, the poor performance of the original method[70] verifies the necessity of noise reduction and optimization schemes in camera self-calibration.

3.6.2 Comparison of Distortion Correction

To verify the effectiveness of our proposed human-tracking-based radial distortion correction, we compare with another method based on the *Manhattan world assumption*, similar to [26, 8, 130]. In the method for comparison, the strong edges are derived from the *Sobel edge detector*, preceded by *Gaussian blur filtering* (see figure 3.11). After filtering away short and weak edges, the strong edges, noted $\{l\}$, are each approximated by second-order polynomial regression. The cost function of the optimization problem is given as

Seq. # - method	Δf (pix.)	Δc_u (pix.)	Δc_v (pix.)	$\Delta\gamma$ (deg.)	$\Delta\beta$ (deg.)	Δt_Z (mm)
1 - ESTHER	**121.5**	*23.3*	**12.7**	*1.64*	*0.39*	**50**
1 - Tang *et al.*[112]	*124.6*	**19.2**	16.0	1.82	1.17	78
1 - Brouwers *et al.*[9]	179.0	43.9	14.8	**1.14**	**0.22**	62
1 - Liu *et al.*[63]	347.0	43.9	*14.8*	N/A	N/A	N/A
1 - Liu *et al.*[64]	229.0	43.9	*14.8*	N/A	N/A	N/A
1 - Wu *et al.*[132]	251.9	43.9	*14.8*	8.68	3.94	N/A
1 - Lv *et al.*[70]	382.7	43.9	*14.8*	15.01	5.47	N/A
2 - ESTHER	**126.5**	**15.1**	*13.7*	*2.61*	1.57	**97**
2 - Tang *et al.*[112]	126.8	19.0	**11.2**	2.90	*1.18*	*115*
2 - Brouwers *et al.*[9]	265.0	41.2	18.0	**0.27**	**0.33**	790
2 - Wu *et al.*[132]	362.0	41.2	18.0	6.45	2.64	N/A
2 - Lv *et al.*[70]	520.3	41.2	18.0	8.93	3.98	N/A
3 - ESTHER	**11.5**	**4.5**	*2.9*	*2.78*	2.07	*116*
3 - Tang *et al.*[112]	*13.1*	*5.3*	**2.8**	3.49	*1.75*	**112**
3 - Brouwers *et al.*[9]	43.0	11.5	9.6	**2.91**	**0.63**	520
3 - Wu *et al.*[132]	28.6	11.5	9.6	7.30	3.04	N/A
3 - Lv *et al.*[70]	34.6	11.5	9.6	11.69	2.07	N/A

Table 3.2: This shows the experimental comparison for camera self-calibration based on walking humans on the benchmarks of VPTZ[95] and EPFL[29]. Bold entries indicate the best results in the corresponding columns for each video sequence, and italics the second-best. Because the estimation of some camera parameters are not considered in several methods, the ground truths are applied for the not applicable values.

Seq. # - method	Δf (pix.)	Δc_u (pix.)	Δc_v (pix.)	$\Delta\gamma$ (deg.)	$\Delta\beta$ (deg.)	Δt_Z (mm)
4 - ESTHER	52.2	*13.8*	6.0	*2.46*	**1.45**	**294**
4 - Tang *et al.*[112]	**51.8**	**12.0**	7.9	**1.84**	*1.75*	*327*
4 - Führ *et al.*[31]	*52.0*	59.8	**5.4**	N/A	N/A	N/A
4 - Wu *et al.*[132]	60.5	59.8	**5.4**	2.77	1.92	N/A
4 - Lv *et al.*[70]	89.6	59.8	**5.4**	7.56	3.29	N/A
5 - ESTHER	**158.5**	*24.9*	*15.9*	*3.17*	**1.89**	*176*
5 - Tang *et al.*[112]	200.1	25.4	16.2	**3.06**	*2.24*	**175**
5 - Führ *et al.*[31]	*197.1*	**0.5**	**0.5**	N/A	N/A	N/A
5 - Wu *et al.*[132]	253.6	**0.5**	**0.5**	4.96	4.17	N/A
5 - Lv *et al.*[70]	280.0	**0.5**	**0.5**	9.41	6.82	N/A

Table 3.3: This shows the experimental comparison for camera self-calibration based on walking humans on the *MOTChallenge* benchmark[50]. Bold entries indicate the best results in the corresponding columns for each video sequence, and italics the second-best. Because the estimation of some camera parameters are not considered in several methods, the ground truths are applied for the not applicable values.

Seq. # - method	Δf (pix.)	Δc_u (pix.)	Δc_v (pix.)	$\Delta\gamma$ (deg.)	$\Delta\beta$ (deg.)	Δt_Z (mm)
6 - ESTHER	**185.2**	**14.6**	**21.1**	**3.14**	**1.26**	**86**
6 - Tang *et al.*[112]	*240.7*	*34.4*	*24.5*	*6.68*	*4.10*	*195*
6 - Wu *et al.*[132]	258.7	62.8	27.2	8.11	5.21	N/A
6 - Lv *et al.*[70]	292.5	62.8	27.2	15.72	8.47	N/A
7 - ESTHER	**191.3**	**22.1**	**17.6**	**2.01**	**1.53**	**121**
7 - Tang *et al.*[112]	*249.5*	*30.5*	38.8	*4.95*	*3.26*	*149*
7 - Wu *et al.*[132]	278.2	61.2	*24.1*	7.91	4.90	N/A
7 - Lv *et al.*[70]	311.5	61.2	*24.1*	15.46	7.41	N/A
8 - ESTHER	**123.3**	**4.9**	**9.6**	**1.80**	**0.92**	**119**
8 - Tang *et al.*[112]	*131.1*	*18.7*	*13.0*	*2.73*	*1.79*	*197*
8 - Wu *et al.*[132]	132.9	43.3	14.5	6.85	2.17	N/A
8 - Lv *et al.*[70]	178.4	43.3	14.5	8.12	4.89	N/A
9 - ESTHER	**219.5**	**16.5**	**19.3**	**2.33**	**1.49**	**162**
9 - Tang *et al.*[112]	260.0	*38.7*	*22.9*	*3.77*	*2.38*	*226*
9 - Wu *et al.*[132]	*258.7*	57.6	30.6	7.19	4.75	N/A
9 - Lv *et al.*[70]	293.3	57.6	30.6	14.83	8.54	N/A

Table 3.4: This shows the experimental comparison for camera self-calibration based on walking humans on our soccer sequences. Bold entries indicate the best results in the corresponding columns for each video sequence, and italics the second-best. Because the estimation of some camera parameters are not considered in several methods, the ground truths are applied for the not applicable values.

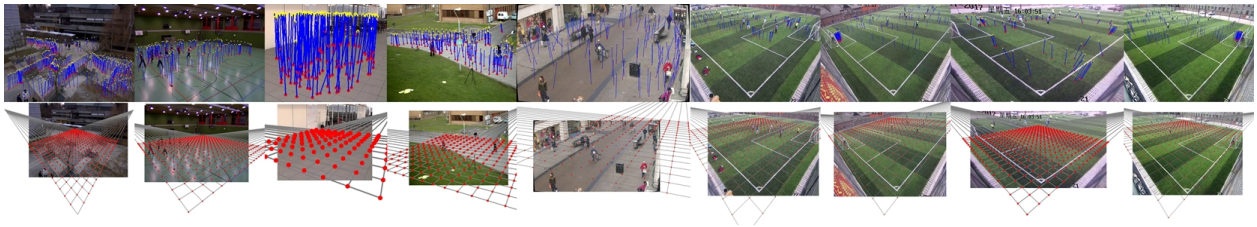

Figure 3.10: This is the visualization of qualitative performance of ESTHER for camera self-calibration on test sequences. The first row shows the located head/foot points. The second row shows the back projected 3D grid on the ground plane. The columns from left to right show Seq. #1, Seq. #2, Seq. #3, Seq. #4, Seq. #5, Seq. #6, Seq. #7, Seq. #8, and Seq. #9.

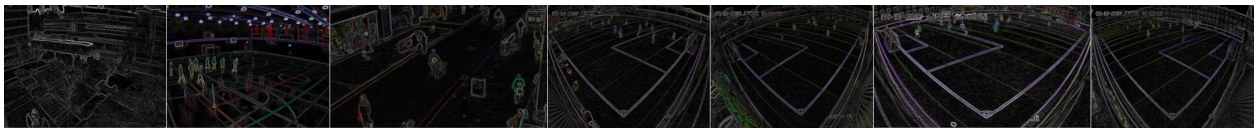

Figure 3.11: These are the edges detected by Sobel edge detector for the method to be compared with in radial distortion correction. The columns from left to right show Seq. #1, Seq. #2, Seq. #5, Seq. #6, Seq. #7, Seq. #8, and Seq. #9.

$$\mathbf{k}^* = \underset{\mathbf{k}\in \mathrm{Rng}_{\mathbf{k}}}{\arg\min} \sum_{l\in\{l\}} \mathrm{curv}(l), \tag{3.20}$$

where curv$(\cdot)$ computes the curvature of an edge segment. In equation 3.20, we search for an optimal set of distortion coefficients that can maximally correct the "curved" straight lines. To solve this problem, we can utilize EDA optimization, whose configuration is the same as algorithm 2.

Experiments are conducted on the seven test sequences with strong radial distortion. The experimental results are summarized in table 3.5, where the implementation based on

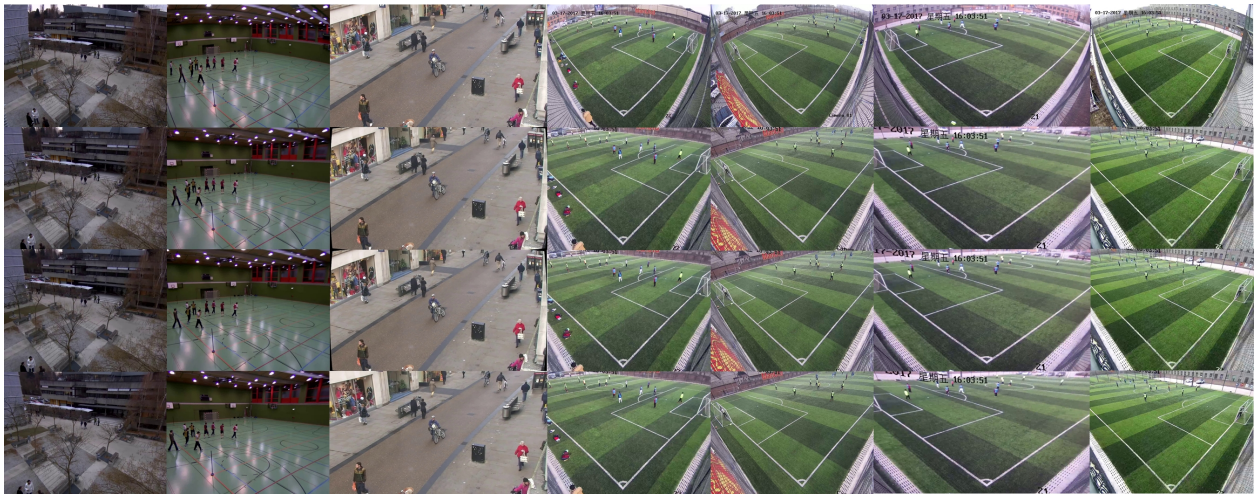

Figure 3.12: This is the qualitative comparison of radial distortion correction on the test sequences. The first row shows the original distorted frame images. The second row shows the images corrected by ground-truth distortion coefficients. The third row shows the images corrected by distortion coefficients estimated by ESTHER. The fourth row shows the images corrected by ESTHER (MWA). The columns from left to right show Seq. #1, Seq. #2, Seq. #5, Seq. #6, Seq. #7, Seq. #8, and Seq. #9.

MWA is denoted as "ESTHER (MWA)". In all the comparisons, the proposed scheme that minimizes human height variance outperforms its opponent. The qualitative performance of the two methods can be visualized in figure 3.12. ESTHER (MWA) tends to overfit the distortion parameters, which is obvious around the frame borders. It is because the detected edges in an image are usually noisy, containing some outliers that are not linear in the undistorted frame image, *e.g.*, branches of trees, some sidelines on the sport courts, *etc.* They cannot be easily filtered away without prior knowledge on the scenes. However, the minimization of relative human height variance is more reliable against noise in most cases, leading to higher accuracy and more natural distortion correction.

Seq. # - method	k_1	k_2
1 - Ground truth	-0.374	0.159
1 - ESTHER	**-0.383**	**0.176**
1 - ESTHER (MWA)	*-0.346*	*0.119*
2 - Ground truth	-0.365	0.131
2 - ESTHER	**-0.327**	**0.117**
2 - ESTHER (MWA)	*-0.479*	*0.198*
5 - Ground truth	-0.602	4.702
5 - ESTHER	**-0.595**	*4.730*
5 - ESTHER (MWA)	*-0.579*	**4.685**
6 - Ground truth	-0.312	0.098
6 - ESTHER	**-0.316**	**0.102**
6 - ESTHER (MWA)	*-0.348*	*0.124*
7 - Ground truth	-0.308	0.101
7 - ESTHER	**-0.322**	**0.107**
7 - ESTHER (MWA)	*-0.351*	*0.119*
8 - Ground truth	-0.469	0.225
8 - ESTHER	**-0.509**	**0.241**
8 - ESTHER (MWA)	*-0.593*	*0.278*
9 - Ground truth	-0.304	0.097
9 - ESTHER	**-0.319**	**0.103**
9 - ESTHER (MWA)	*-0.346*	*0.119*

Table 3.5: This is the experimental comparison for radial distortion correction on test sequences. Bold entries indicate the best results in the corresponding columns for each video sequence, and italics the second-best.

dist. opt.	cam. opt.	L_∞ est.	V_∞ est.	Δf (pix.)	Δc_u (pix.)	Δc_v (pix.)	$\Delta\gamma$ (deg.)	$\Delta\beta$ (deg.)	Δt_Z (mm)	Δk_1	Δk_2
EDA	EDA	LLR	MSC	**121.5**	*23.3*	**12.7**	*1.64*	**0.39**	*50*	**0.009**	**0.017**
LM	EDA	LLR	MSC	128.4	26.8	*13.0*	**1.27**	0.94	**47**	0.058	0.032
EDA	LM	LLR	MSC	129.7	31.2	17.5	1.85	1.11	58	*0.030*	*0.021*
LM	LM	LLR	MSC	132.9	33.4	16.3	2.03	*0.72*	72	0.049	0.029
N/A	EDA	LLR	MSC	*124.6*	**19.2**	16.0	1.82	1.17	78	N/A	N/A
EDA	N/A	LLR	MSC	167.3	34.1	16.2	3.94	2.94	N/A	0.051	0.042
N/A	N/A	LLR	MSC	185.1	43.9	14.8	5.06	3.26	N/A	N/A	N/A
N/A	N/A	RANSAC	MSC	207.5	43.9	14.8	6.22	3.63	N/A	N/A	N/A
N/A	N/A	LLR	RANSAC	261.6	43.9	14.8	8.99	3.46	N/A	N/A	N/A
N/A	N/A	RANSAC	RANSAC	351.9	43.9	14.8	8.68	3.94	N/A	N/A	N/A
N/A	N/A	N/A	MSC	409.3	43.9	14.8	9.69	4.32	N/A	N/A	N/A
N/A	N/A	LLR	N/A	430.0	43.9	14.8	9.95	4.28	N/A	N/A	N/A
N/A	N/A	N/A	N/A	382.7	43.9	14.8	15.01	5.47	N/A	N/A	N/A

Table 3.6: This is the ablation study of ESTHER on the *Outdoor* sequence from the VPTZ benchmark[95].

3.6.3 Ablation Study

We further study the effect of each individual algorithmic component. For ablation study, we adopt the *Outdoor* sequence, *i.e.*, Seq. #1, in our experiments. The experimental results are presented in table 3.6 and table 3.7, where "LLR" and "MSC" respectively stand for Laplace linear regression and mean shift clustering. We not only compare with the scenarios where some of the modules are missing, but also the cases when EDA is substituted by the LM algorithm for optimization and/or RANSAC is adopted for the estimation of vanishing points. All the experiments are conducted under the same configuration setting.

In table 3.6, we can observe that all the methods with either EDA or LM optimization show significant improvement in estimation accuracy, as the extra information from the

dist. opt.	cam. opt.	L_∞ est.	V_∞ est.	reprojection error (pix.)	relative human height standard deviation (%)
EDA	EDA	LLR	MSC	**1.06e-3**	*1.47*
LM	EDA	LLR	MSC	*2.06e-1*	1.92
EDA	LM	LLR	MSC	2.95e-3	**1.43**
LM	LM	LLR	MSC	7.17e-1	2.05
N/A	EDA	LLR	MSC	5.69e-3	6.09
EDA	N/A	LLR	MSC	3.69	1.90
N/A	N/A	LLR	MSC	4.65	5.81
N/A	N/A	RANSAC	MSC	3.12	4.97
N/A	N/A	LLR	RANSAC	5.00	8.02
N/A	N/A	RANSAC	RANSAC	4.21	5.91
N/A	N/A	N/A	MSC	8.10	6.40
N/A	N/A	LLR	N/A	5.83	5.79
N/A	N/A	N/A	N/A	6.11	7.13

Table 3.7: This is the ablation study of ESTHER in terms of final cost values on the *Outdoor* sequence from the VPTZ benchmark[95].

scene and video objects is exploited in the minimization of cost functions. Moreover, the constraints on unknown intrinsic camera parameters are relaxed. In table 3.7, both EDA and LM algorithm can successfully minimize the cost values, but the effectiveness of evolutionary optimization is superior. It is because LM optimization is based on stochastic gradient descent, which starts searching from local region. But evolutionary algorithm directly operates on the global solution domain, and thus can better avoid local optima.

The noise reduction in L_∞ and V_∞ estimation is key to the optimization of camera parameters, as accurate vanishing points usually generate good initial values for optimization. In table 3.6, we can also learn the enhanced robustness by Laplace linear regression and mean shift clustering. In the estimation of L_∞, the line fitting based on probabilistic modeling is more reliable than RANSAC, as the entire set of data points is exploited. As for V_∞ estimation, since the shape of the cluster for inliers can be tightly defined in mean shift clustering, our proposed scheme also outperforms RANSAC.

3.6.4 Computational Complexity Analysis

The computational complexity analysis of our proposed algorithm is provided as follows. Assume that the number of collected object instances is N. In radial distortion correction, the relative human height variance is computed at every EDA generation, so the computation time is $\mathrm{O}(N)$. As for the implementation based on MWA, the computation time is $\mathrm{O}(\#(l))$. Though the number of edge segments in a frame image is usually smaller than the number of human instances in a video sequence, the optimization performance can be highly sensitive to the quality of the selected frame image. As for the optimization of camera parameters, compared to other approaches using nonlinear optimization based on human height distribution[71, 31], whose computation time is $\mathrm{O}(N)$, our formulation based on reprojection error on the ground plane only depends on the size of the pre-defined 3D grid points. Thus, the computation time is only $\mathrm{O}(1)$. In mean shift clustering, we visit every candidate point once, each generated by a pair of human instances, so that the computation time is $\mathrm{O}(N^2)$, which is the *best conceivable runtime* (BCR). This may be slower than RANSAC

based on random sampling, but we exploit all the useful information. Finally, in Laplace linear regression, the runtime is also O(N^2).

The proposed framework is implemented in C++ with the support of the OpenCV 3 library. It is run on an Intel Core i7-7700HQ PC with 4 cores, 2.80 GHz processor and 16 GB RAM in the Windows 10 environment. To ensure fast computation, we start self-calibration and radial distortion correction once the numbers of candidate points for L_∞ and V_∞ estimation both exceed 1,000. After head/foot localization, the algorithmic process takes 48.7 seconds to complete.

3.6.5 Application to 3D Object Tracking

An intuitive application of camera self-calibration is to automatically back project the 2D tracking into 3D space. Here we demonstrate the utilization of ESTHER in 3D MOT.

Both test sequences, *PETS09-S2L1* and *AVG-TownCentre*, are included in the *MOTChallenge* 3D benchmark[50] for evaluating single-camera MOT. They have also been adopted for experiments in the work by Führ *et al.*[31], who test the self-calibration strategy with their own 3D tracking algorithm[30]. In our work, the camera projection matrix estimated by ESTHER is applied to the state-of-the-art tracking-by-detection method on the benchmark, MOANA. In table 3.8, we present the experimental results of object tracking, where the metric we use is MOTA[5]. The accurate back projection of object instances to 3D space by ESTHER largely improves the tracking accuracy of MOANA. Though the nonlinear optimization by Führ *et al.*[31] can significantly improve their initial estimation, their tracking accuracy is still inferior to the proposed method. Finally, we also compare with another state-of-the-art, DP+NMS[93], and the baseline by Leal-Taixé *et al.*[50]. Their tracking predictions are also less reliable than ours. The demo video sequences are available on the website of *MOTChallenge*. [1]

[1]The demo videos of the test sequences of the *MOTChallenge* 3D benchmark are available at https://motchallenge.net/vis.

Method	*PETS09-S2L1*	*AVG-TownCentre*
3D MOANA + ESTHER	**81.5**	**46.1**
2D MOANA	75.2	41.8
Führ *et al.*[31] opt.	55.3	*44.9*
Führ *et al.*[31] init.	51.4	19.9
DP+NMS[93]	58.1	38.0
Baseline[50]	*77.5*	35.9

Table 3.8: This is the experimental comparison of single-camera tracking in MOTA on the *MOTChallenge* 3D benchmark[50].

Chapter 4

MODELING OF OBJECT APPEARANCE BY NORMALIZED ADAPTATION

We present in this chapter our work on adaptive modeling of object appearance for robust 3D tracking, which covers parts of our publications[108, 115, 109]. The overview flow diagram of our proposed framework is shown in figure 4.1. We first exploit the output of preliminary 2D human tracking and foreground segmentation by MAST for camera self-calibration using ESTHER. The observations of objects can be located by object detection or with the assistance of segmentation. When a target is not occluded by or grouped with other object(s), it is associated with available observation(s) based on an efficient 3D Kalman-filter-based strategy. The proposed appearance models and a probabilistic model of 3D object properties are learned online. When an observation is grouped with others, the cross-matching module is enabled to associate nearby targets based on the unoccluded area of appearance models. On the other hand, when an object is seriously occluded or missing, his/her appearance model is temporarily stored and used for re-identification. The detailed formulation and the role of each component are illustrated as follows.

4.1 Formulation of Data Association

Before introducing the proposed adaptive appearance model for object tracking, we first define the formulation of MOT as a data association problem in time and space. We aim to recover the trajectories $\mathcal{T}$ of all targets within the 3D scene, which are defined as

$$\mathcal{T} = \{\mathcal{T}_i : i = 1, 2, \ldots, |\mathcal{T}|\}, \tag{4.1}$$

where each $\mathcal{T}_i$ is equivalent to an object identity.

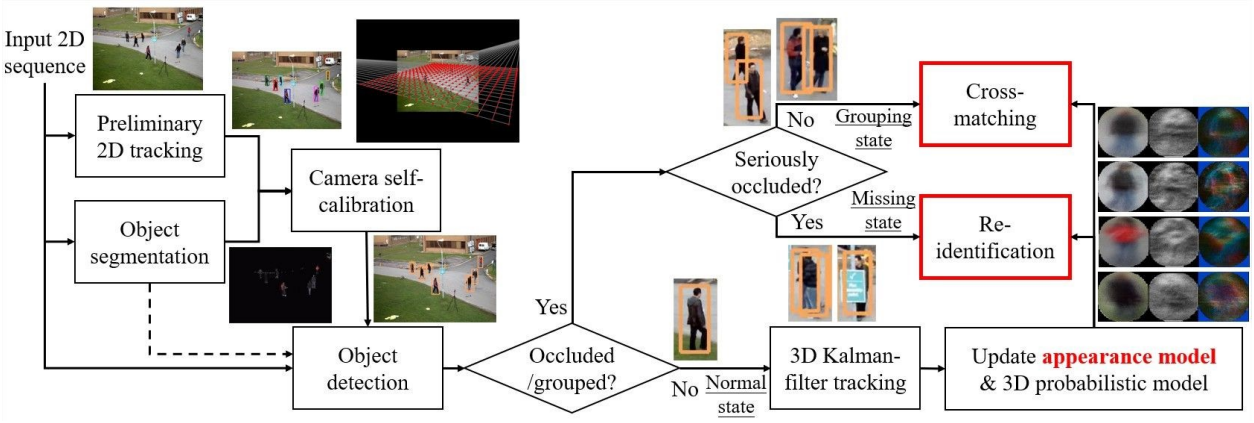

Figure 4.1: This is the flow diagram of MOANA for adaptive modeling of object appearance to support robust 3D tracking.

The basic units of MOT are the candidate observations of objects, noted $\mathcal{O}$, derived from object detection or with the assistance of foreground segmentation, defined as

$$\mathcal{O} = \{\mathcal{O}_j \sim (g_j, f_j, q_j, t_j) : j = 1, 2, \ldots, |\mathcal{O}|\}, \tag{4.2}$$

in which g_j is the 3D geometry information, f_j is the extracted appearance feature, q_j is the foreground mask within the object region, and t_j is the time stamp. They will be illustrated in detail in the following sections.

The goal of MOT is to solve the following objective from an input video sequence

$$\mathcal{T}_i \leftarrow \mathcal{O}_j, \forall i, \forall j, \tag{4.3}$$

which represents the assignment of every observation to a corresponding object identity. The false positives are all assigned to $\mathcal{T}_\infty$.

When the camera parameters are unavailable, we first process a short period of the video sequence by preliminary 2D tracking and foreground segmentation using MAST. Each human object is modeled as a pole perpendicular to the ground plane, whose endpoints

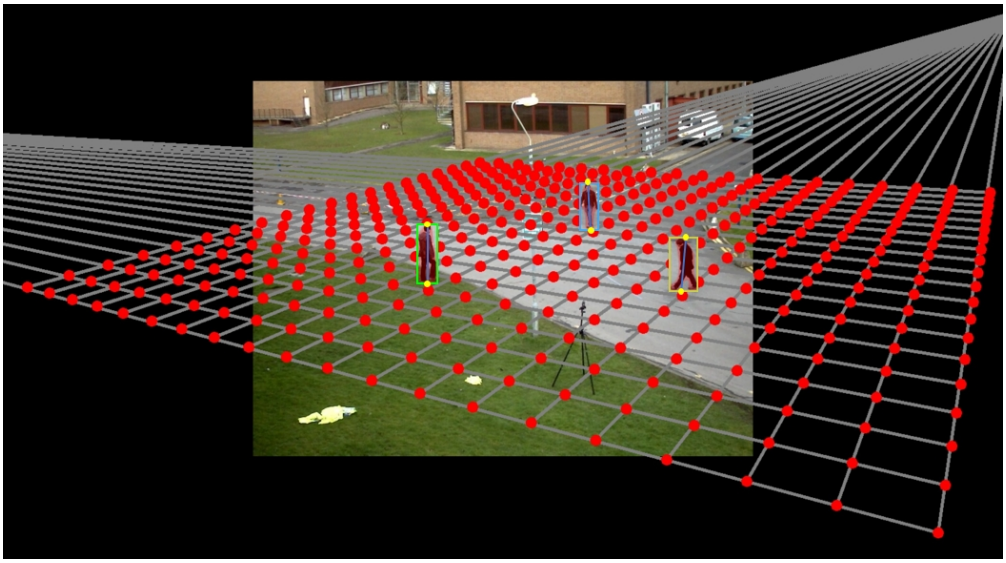

Figure 4.2: This is the projected 3D grid on the ground plane generated by camera self-calibration with the extracted head and foot points highlighted.

are located based on the orientation of the foreground blob, from which we can compute the horizon line and vanishing points in the scene for camera self-calibration by applying ESTHER. An example of the estimated 3D ground plane from camera self-calibration is shown in figure 4.2. Then, we process the sequence from the beginning with each object observation back projected to 3D space. The geometry information of each $\mathcal{O}_j$, noted g_j, is represented by six aspects.

$$g_j \sim (b_j, P_j, D_j, V_j, W_j, H_j), \tag{4.4}$$

where $b_j \in \mathbb{R}^4$ denotes the 2D bounding box represented in terms of centroid coordinates and size, $P_j \in \mathbb{R}^2$ denotes the back projected foot point coordinates on the 3D ground plane, *i.e.*, the X-Y plane, D_j denotes the 3D depth of P_j, $V_j \in \mathbb{R}^2$ denotes the 3D velocity of P_j on the X-Y plane, and W_j and H_j are the width and height of the 3D bounding box, respectively.

An observation $\mathcal{O}_j$ is deemed to be under occlusion or grouped with other object(s) if b_j

overlaps with other(s) or the 3D distance of their foot points is smaller than a threshold τ_P. Otherwise, $\mathcal{O}_j$ is associated with a $\mathcal{T}_i$ based on an efficient 3D Kalman-filter-based approach. The state vector of the Kalman filter has six dimensions, corresponding to $P_j \in \mathbb{R}^2$, $V_j \in \mathbb{R}^2$, W_j and H_j, whose prediction and update are similar to the 2D scenario[18].

The Kalman prediction of a target $\mathcal{T}_i$ is regarded as a predicted observation, noted $\hat{\mathcal{O}}_i$. An observation $\mathcal{O}_j$ is associated with $\mathcal{T}_i$ based on the following rule

$$\mathcal{T}_i \leftarrow \mathcal{O}_j, \quad \text{if} \quad \frac{\|\hat{P}_i - P_j\|_2}{w_j} < \tau_P, \tag{4.5}$$

which means that the predicted 3D foot point $\hat{P}_i$ of $\mathcal{T}_i$ is within a short Euclidean distance of $\mathcal{O}_j$. The term w_j is proportional to the depth of $\mathcal{O}_j$, defined by

$$w_j = D_j \cdot \eta_D + c_D, \tag{4.6}$$

where η_D is a constant step size and the addition of a constant c_D is to avoid division-by-zero error. The division by w_j in equation 4.5 is to compensate for the ambiguity in 3D measurement of distant objects, whose estimated 3D foot points are highly sensitive to small errors in object detection and/or foreground segmentation.

When tracking under the mode of Kalman filtering, we also build a probabilistic model of 3D object properties online. The probabilistic model has four dimensions, corresponding to $V_j \in \mathbb{R}^2$, W_j and H_j. A four-dimension probabilistic model is used to actively learn the normal distribution of each 3D property. False positives of object observations are removed from the list of candidates for association based on the three-sigma rule of thumb in normal distribution.

4.2 Adaptive Modeling of Object Appearance

Even though 3D Kalman-filter-based tracking can generate more reliable tracklets compared to 2D tracking, it still cannot overcome the problem of identity switch during interaction between objects. To resolve the ambiguity between objects that are spatially close to each

other, we propose an adaptive model to learn the change of object appearance online. The appearance model of a target $\mathcal{T}_i$, noted $\mathbf{m}_i$, is a combination of d sub-models, where d is the feature dimension. Each sub-model contains a set of n observed feature values.

$$\mathbf{m}_i = \{m_i^1(p), m_i^2(p), \ldots, m_i^n(p) | \forall p \in [1, d]\}, \tag{4.7}$$

The procedure of model construction and update is demonstrated in figure 4.3. In this example, the features are extracted from normalized pixel templates of size $d = w \times h$. The dimension of each feature vector is given by $m_i^k(p) \in \mathbb{R}^6$, as it encodes RGB values in 3 channels, LBP values in 1 channel, as well as gradient magnitudes and angles. To initialize or update this appearance model, each pixel template within the object region is normalized to the size of $w \times h$ (see the top right of figure 4.3). As shown in the bottom left of figure 4.3, The foreground mask q_j is used to determine the visible object region. When the observation is occluded, the occluded area is eliminated from q_j. The update rate of each sub-model, noted $\alpha_i(p)$, is dynamically controlled by a softmax function, which depends on the distance between newly observed features and values in the past. We define $\alpha_i(p)$ as follows

$$\alpha_i(p) = (1 + \exp[\min_k \|f_j(p) - m_i^k(p)\|_2 - \tau_f])^{-1}, \tag{4.8}$$

where $f_j(p)$ is the newly observed feature vector of the same dimension as $m_i^k(p)$. The term τ_f is the maximum distance threshold in the feature space. New features that vary from the past are more likely to be updated, as they reflect the change of appearance that should be learned.

For pixel-based features like the examples in figure 4.3, a Gaussian spatial weighting scheme is also employed to adjust the learning rate as

$$\alpha_i(p) = \frac{\exp[-\frac{\|p - p_c\|_2^2}{2(w^2+h^2)}]}{1 + \exp[\min_k \|f_j(p) - m_i^k(p)\|_2 - \tau_f]}, \tag{4.9}$$

where p_c denotes the center of mass of the visible area within the object region. The spatially weighted learning rates $\alpha_i(p)$ are maximum around the central region, which the body of the

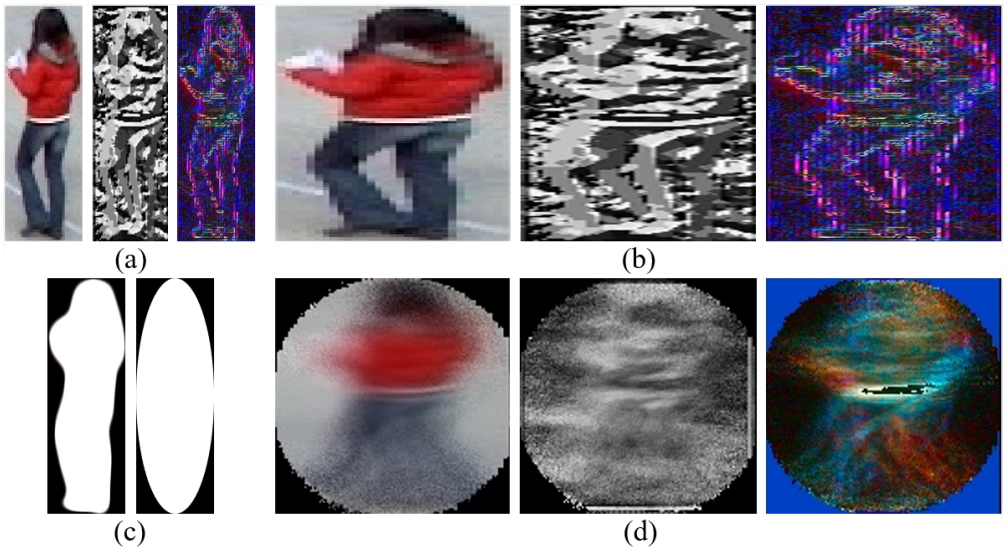

Figure 4.3: This is an example of the construction and update of MOANA. (a) The RGB image for color representation, the LBP image for texture representation and the gradient image for edge representation. (b) Feature maps normalized to $w \times h$. (c) The foreground masks used to indicate visible object area to be updated. When the segmentation results are not available, a maximum-ellipse mask is used. (d) The visualization of the averaged feature components in the adaptive appearance model.

object usually occupies, so the sub-models there should be updated more frequently. The learning rate drops as p gets further away from p_c. Thus, we can suppress the influence of background area.

The procedure of model update is described as follows. When a candidate observation $\mathcal{O}_j$ is associated with $\mathcal{T}_i$ in Kalman filtering, the extracted features f_j are used to update the appearance model of $\mathcal{T}_i$, *i.e.*, $\mathbf{m}_i$. For each sub-model in $\mathbf{m}_i$, if there are less than n feature vectors stored, the observed feature vector $f_j(p)$ is added into the sub-model by a probability of $\alpha_i(p)$. Otherwise, a random feature vector $m_i^k(p)$ in the sub-model is swapped by $f_j(p)$ with a probability of $\alpha_i(p)$. At the bottom right of figure 4.3, each feature component of

MOANA is plotted, in which averaged values are displayed.

To measure the appearance affinity using the proposed model, the similarity score between the prediction of $\mathcal{T}_i$, noted $\hat{\mathcal{O}}_i$, and an observation $\mathcal{O}_j$ is given as

$$s(\hat{\mathcal{O}}_i, \mathcal{O}_j) = \frac{\sum_k [\#(\|f_j(p) - m_i^k(p)\|_2 < \tau_f, \forall k \leq n)]}{dn}, \tag{4.10}$$

where $\#(\cdot)$ returns the number of samples satisfying the given condition. The value of $s(\hat{\mathcal{O}}_i, \mathcal{O}_j)$ is between 0 and 1, where higher value indicates higher similarity, because more features are matched between the prediction and the observation.

Note that the proposed appearance model is universal, *i.e.*, compatible with all kinds of feature combinations, as long as the feature dimension is fixed. Thus, in the example of figure 4.3, all the pixel templates need to be normalized to $w \times h$. MOANA is also compatible with different measurements of distance in the feature space. Besides, the computation of model update and comparison is always constant, *i.e.*, O(dn). With reasonable setting of configuration parameters, the processing speed can be sufficiently fast to support real-time application. Moreover, different from previous approaches, since a set of previously observed feature values is stored and updated in random, MOANA is capable of "memorizing" a relatively long-term history of appearance change, which may cover different viewing angles, object poses and illumination. The proposed method also benefits from the normalized similarity score between 0 and 1, which makes it convenient to set thresholds and compare with each other. On the other hand, common affinity measurements, such as Bhattacharya distance and KL divergence, do not share such property.

4.3 Cross-matching with Appearance Model

The cross-matching module is enabled when a candidate observation is spatially close to other object(s) but has more than 50% of the object region visible, *i.e.*, under the grouping state. In this case, a predicted target location by Kalman filter may be associated with a wrong observation easily, which leads to identity switch. The problem can be mitigated by comparing the appearance features across grouped objects, *i.e.*, cross-matching, but the

effect is limited when the nearby targets share high appearance similarity. Since long-term appearance change is effectively encoded in our proposed appearance model, we can maximally distinguish highly similar objects through cross-matching.

The procedure of cross-matching is demonstrated in figure 4.4. More specifically, for each observation $\mathcal{O}_j$ a list of nearby target predictions, noted $\mathbf{l}_j$, is kept. If there are more than one prediction in $\mathbf{l}_j$, $\mathcal{O}_j$ is in the grouping state. In cross-matching, the observation $\mathcal{O}_j$ is compared with each element in $\mathbf{l}_j$. The computation of similarity score incorporates both 3D geometry information and appearance affinity, defined as

$$s_c(\hat{\mathcal{O}}_i, \mathcal{O}_j) = s(\hat{\mathcal{O}}_i, \mathcal{O}_j) \cdot \frac{w_j}{\|\hat{P}_i - P_j\|_2}, \hat{\mathcal{O}}_i \in \mathbf{l}_j, \tag{4.11}$$

where the subscript c refers to cross-matching. The similarity score in equation 4.10 is divided by the Euclidean distance of 3D foot points, because spatially close objects are more likely to be associated. Similar to equation 4.5, the term w_j is added to compensate for the ambiguity of foot point estimation of distant objects. With the set of computed scores $\{s_c(\hat{\mathcal{O}}_i, \mathcal{O}_j)\}$ between each pair of observation and prediction, we formulate a bipartite matching problem, which can be effectively solved using the *Hungarian algorithm*. The detailed pseudocode of the above procedure is provided in algorithm 3.

4.4 Re-identification with Appearance Model

When an object observation is under serious occlusion, *i.e.*, the visible area is smaller than 50% or there is no nearby target prediction (false negative), his/her leaving time stamp, location, and appearance model are temporarily stored for re-identification. Since the viewpoint of a target usually changes significantly after serious occlusion, and targets frequently enter and exit the *region of interest* (ROI) in real world, a reliable appearance descriptor that learns long-term appearance variation is key to the success of re-identification.

The procedure of re-identification is demonstrated in figure 4.5. For each entering observation $\mathcal{O}_j$ that is not associated with any existing target, it is compared with a list of

Algorithm 3 Cross-matching based on MOANA

Require: current video frame, candidate observations in the input frame $\{\mathcal{O}_j\}$, prediction of each target from the Kalman filter $\{\hat{\mathcal{O}}_i\}$

Ensure: matched pairs of predictions and observations $(\forall \mathcal{T}_i \leftarrow \mathcal{O}_j)$

for all $\mathcal{O}_j \in \{\mathcal{O}_j\}$ **do**
  clear the list of nearby candidate predictions $\mathbf{l}_j$;
  for all $\hat{\mathcal{O}}_i \in \{\hat{\mathcal{O}}_i\}$ **do**
    if ($\frac{\|\hat{P}_i - P_j\|_2}{w_j} < \tau_P$ or $\hat{b_i}$ overlaps with b_j) and $\frac{\text{visible area of } b_j}{\text{total area of } b_j} > 50\%$ **then**
      push $\hat{\mathcal{O}}_i$ into $\mathbf{l}_j$;
    end if
  end for
  if $\#(\hat{\mathcal{O}}_i \in \mathbf{l}_j) > 1$ **then**
    for all $\hat{\mathcal{O}}_i \in \mathbf{l}_j$ **do**
      compute $s_c(\hat{\mathcal{O}}_i, \mathcal{O}_j)$ using equation 4.11;
      push $s_c(\hat{\mathcal{O}}_i, \mathcal{O}_j)$ into $\{s_c(\hat{\mathcal{O}}_i, \mathcal{O}_j)\}$;
    end for
  end if
end for
solve the association based on $\{s_c(\hat{\mathcal{O}}_i, \mathcal{O}_j)\}$ using the Hungarian algorithm;
output all matched pairs of $\mathcal{T}_i$ and $\mathcal{O}_j$.

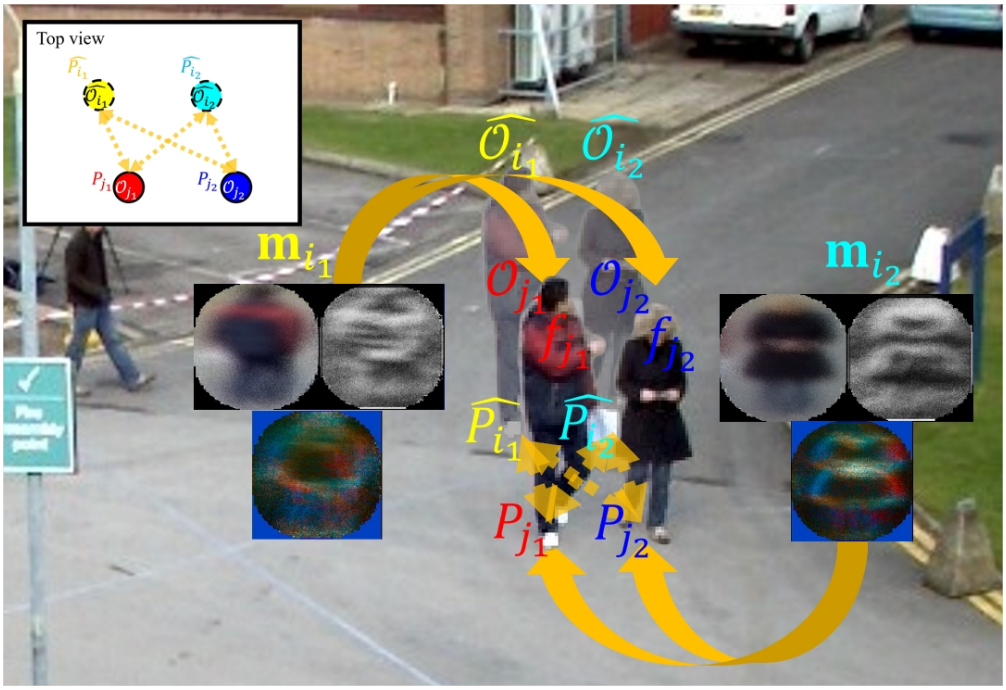

Figure 4.4: This is the demonstration of cross-matching for observations grouped with each other, based on 3D geometry information and the proposed adaptive appearance model.

disappeared targets, noted $\mathcal{T}'$. If a missing target is successfully associated with an entering observation, its identity and appearance model is recovered. The similarity score for re-identification is computed as

$$s_r(\hat{\mathcal{O}}_i, \mathcal{O}_j) = \begin{cases} s(\hat{\mathcal{O}}'_i, \mathcal{O}_j) \cdot \frac{(t_j - t'_i) \cdot w_j^D}{\|\hat{P}'_i - P_j\|_2}, & \text{if} \quad \frac{\|\hat{P}'_i - P_j\|_2}{(t_j - t'_i) \cdot w_j} < \tau_P \\ 0, & \text{otherwise} \end{cases}, \tag{4.12}$$

in which the subscript r stands for re-identification. $\hat{\mathcal{O}}_i$ is the Kalman prediction of a missing target at t_j. $\hat{P}'_i$ and t'_i are the predicted 3D location at the current frame and the time stamp that the target disappears respectively. Different from equation 4.12, we have a new term, $(t_j - t'_i)$, which calculates the time span in second that the target has been missing. It is intuitive that a target missing for a long time usually leads to higher uncertainty in

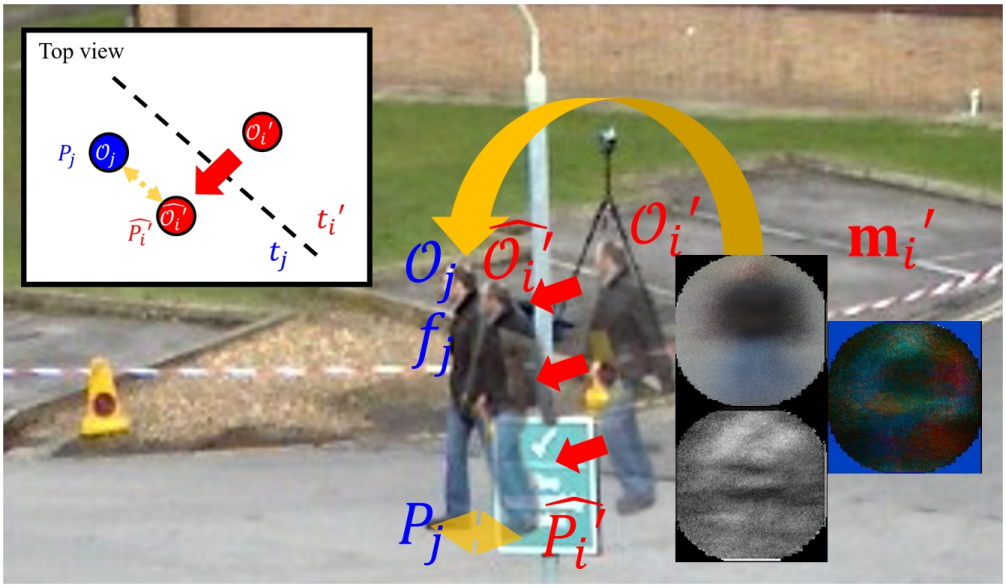

Figure 4.5: This is the demonstration of re-identification for missing observations, based on 3D geometry information and the proposed adaptive appearance model. A disappeared object, $\mathcal{T}_i'$, is shifted to a predicted location, *i.e.*, $\hat{\mathcal{O}}_i'$, and compared with an entering observation $\mathcal{O}_j$.

the prediction of location. Moreover, only the prediction(s) in the neighborhood of the observation are considered as candidate(s) for re-identification. Finally, if the observation is not associated with any missing target or the maximum similarity score is considered too low, *i.e.*, smaller than a threshold noted τ_s, it is identified as a new target. The pseudocode of re-identification is detailed in algorithm 4.

4.5 Experimental Results

Extensive experiments are conducted on the publicly available *MOTChallenge* benchmark[50], which is a collection of existing and new data for MOT evaluation. It is developed to bring forward the strengths and weaknesses of the state-of-the-art MOT methods. Among all the subsets of this benchmark[50, 75], *MOTChallenge* 2015 3D is the only one dedicated for the

Algorithm 4 Re-identification based on MOANA

Require: current video frame, entering observations in the input frame $\{\mathcal{O}_j\}$, prediction of each disappeared target at the current frame $\{\hat{\mathcal{O}}'_i\}$

Ensure: object identities of $\{\mathcal{O}_j\}$ $(\forall \mathcal{T}_i \leftarrow \mathcal{O}_j)$

for all $\mathcal{O}_j \in \{\mathcal{O}_j\}$ **do**
  for all $\hat{\mathcal{O}}'_i \in \{\hat{\mathcal{O}}'_i\}$ **do**
    if $\frac{\|\hat{P}'_i - P_j\|_2}{(t_j - t'_i)\cdot w_j} < \tau_P$ **then**
      compute $s_r(\hat{\mathcal{O}}'_i, \mathcal{O}_j)$ using equation 4.12;
      push $s_r(\hat{\mathcal{O}}'_i, \mathcal{O}_j)$ into $\{s_r(\hat{\mathcal{O}}'_i, \mathcal{O}_j)\}$;
    end if
  end for
  $\hat{\mathcal{O}}'^{*}_i \leftarrow \arg\max_{\forall \hat{\mathcal{O}}'_i \in \{\hat{\mathcal{O}}'_i\}} \{s_r(\hat{\mathcal{O}}'_i, \mathcal{O}_j)\}$
  if $s_r(\hat{\mathcal{O}}'^{*}_i, \mathcal{O}_j) > \tau_s$ **then**
    assign the identity of $\mathcal{T}'^{*}_i$ to $\mathcal{O}_j$;
  else
    assign a new identity to $\mathcal{O}_j$;
  end if
end for
output all identities of $\{\mathcal{O}_j\}$.

evaluation of 3D tracking performance, in which all the videos are taken by static cameras, so that our camera self-calibration scheme can be applied. There are two training sequences, *PETS09-S2L1* and *TUD-Stadtmitte*, with 974 frames and 5,632 ground-truth bounding boxes for 29 targets in total. *AVG-TownCentre* and *PETS09-S2L2* are the two test sequences, which are significantly more complex than the training set, including 886 frames and 16,789 ground truths for 268 targets. The benchmark presents all kinds of evaluation metrics for the performance of MOT[5], such as MOTA, *multiple object tracking precision* (MOTP), *false positives* (FP), *false negatives* (FN), *identity switches* (ID Sw.), *mostly tracked targets* (MT), *mostly lost targets* (ML), *fragments* (Frag.), *etc.* The definitions of all the evaluation metrics are summarized in table 4.1. The two test sequences in the *MOTChallenge* 3D benchmark are included in the *MOTChallenge* 2D benchmark as well, which also allow us to compare with the state-of-art in 2D MOT[13, 100, 43, 137, 19, 124, 103, 16, 143].

The proposed framework is implemented in C++ with the support of the OpenCV 3 library. It is run on an Intel Core i7-7700K PC with 4 cores, 4.20 GHz processor and 24 GB RAM in the Ubuntu 14.04 environment. After testing different features including pixel templates, histograms, deep learning features and their combinations on the training sequences, we choose to incorporate both RGB and LBP pixel templates in our appearance model for the evaluation on the test sequences. The distance measurement in feature space is given by the Euclidean distance. The minimal color distance threshold and minimal LBP distance threshold, *i.e.*, τ_f, are both empirically set to 30. In all the experimental sequences, the normalized size for feature extraction is empirically set as $w \times h = 64 \times 64$, which is an ideal balance between HD resolution and real-time computation. Due to the relatively short-term appearance of most objects in these sequences, n is set to 3 seconds. In addition, the values of τ_P, τ_s, η_D and c_D are empirically chosen to be 2 meters, 0.30, 1/30 and 1 respectively. Moreover, the gaps between re-identified tracklets are linearly interpolated. To conform with the provided ground truth of camera parameters, we compute the transformation from the estimated projection matrix to the actual homography, so that our 3D tracking results can be converted properly for evaluation. The unit used for all 3D measurements is meter.

Measure	Better	Perfect	Description
Avg Rank	↓	1	This is the rank of each tracker averaged over all present evaluation measures.
MOTA	↑	100%	Multiple Object Tracking Accuracy. This measure combines three error sources: false positives, missed targets and identity switches.
MOTP	↑	100%	Multiple Object Tracking Precision. The misalignment between the annotated and the predicted object locations.
MT	↑	100%	Mostly tracked targets. The ratio of ground-truth trajectories that are covered by a track hypothesis for at least 80% of their respective life span.
ML	↓	0%	Mostly lost targets. The ratio of ground-truth trajectories that are covered by a track hypothesis for at most 20% of their respective life span.
FP	↓	0	The total number of false positives.
FN	↓	0	The total number of false negatives (missed targets).
ID Sw.	↓	0	The total number of identity switches.
Frag	↓	0	The total number of times a trajectory is fragmented (*i.e.*, interrupted during tracking).
Hz	↑	∞	Processing speed (in frames per second) on the benchmark.

Table 4.1: This is the summary of the evaluation metrics used in *MOTChallenge*.

Tracker	Avg Rank	MOTA	MOTP	MT	ML	FP	FN	ID Sw.	Frag	Hz
MOANA	**3.2**	**52.7**	56.3	28.4	22.0	2,226	**5,551**	**167**	586	19.4
DBN	3.4	51.1	61.0	**28.7**	17.9	2,077	5,746	380	418	0.1
GPDBN	3.4	49.8	**62.2**	25.7	17.2	1,813	6,300	311	**386**	0.1
GustavHX	3.8	42.5	56.2	25.7	15.7	2,735	6,623	302	431	0.0
MCFPHD	4.8	39.9	53.6	25.7	16.8	3,029	6,700	363	529	17.7
MCG	6.2	35.9	54.8	8.2	25.7	**1,600**	8,464	692	1,017	0.1
LPSFM	5.2	35.9	54.0	13.8	21.6	2,031	8,206	520	601	8.4
LP3D	4.9	35.9	53.3	20.9	16.4	3,588	6,593	580	659	**83.5**
SVT	6.8	34.2	55.8	11.2	25.4	3,057	7,454	532	611	1.9
AMIR3D	7.1	25.0	55.6	3.0	27.6	2,038	9,084	1,462	1,647	1.2
KalmanSFM	6.3	25.0	53.6	6.7	**14.6**	3,161	7,599	1,838	1,686	30.6

Table 4.2: This is the comparison of the proposed method with the state-of-the-art on the *MOTChallenge* 3D benchmark (test sequences). Bold entries indicate the best results in the corresponding columns.

4.5.1 Comparison with the State-of-the-art in 3D

Currently, there have been 11 submissions on the *MOTChallenge* 2015 3D benchmark, including two anonymous methods. All the experimental results are summarized in table 4.2. The corresponding qualitative visualization is available in figure 4.6. The demo videos can be viewed on the *MOTChallenge* website. [1] Note that the noisy detection sets provided by the benchmark are used as input to our algorithm. To be fair with other methods in comparison, we do not apply foreground segmentation in our appearance model. Thus, each object mask in the proposed appearance model is defined by a maximum ellipse (see figure 4.3).

MOANA is currently ranked on top in terms of the two most significant metrics, MOTA

[1]The demo videos of the test sequences of the *MOTChallenge* 3D benchmark are available at https://motchallenge.net/vis/PETS09-S2L2/MOANA and https://motchallenge.net/vis/AVG-TownCentre/MOANA.

Figure 4.6: This is the qualitative comparison on the test sequences of the *MOTChallenge* 3D benchmark, which can be better visualized through demo videos on the *MOTChallenge* website. First row: Frame #91 of *PETS09-S2L2*. Second row: Frame #222 of *PETS09-S2L2*. Third row: Frame #409 of *PETS09-S2L2*. Fourth row: Frame #128 of *AVG-TownCentre*. Fifth row: Frame #189 of *AVG-TownCentre*. Sixth row: Frame #441 of *AVG-TownCentre*. First column: MOANA. Second column: DBN[44]. Third column: MCFPHD[129]. Fourth column: LPSFM[51]. Fifth column: LP3D[50]. Sixth column: KalmanSFM[92].

and ID Sw. As shown in figure 4.6 and the online demo videos, our predicted trajectories and localization of targets are all relatively more accurate, whereas other methods miss a few more targets and introduc more false positives. The promising performance mainly benefits from the proposed appearance adaptation scheme that maintains robustness against occlusion and appearance similarity among nearby targets. This is proven by the fact that our ID Sw. on this challenging benchmark is reduced by over 46% compared with the former leader. This also explains why MOANA enjoys a relatively high MT score. However, a

drawback of our interpolation scheme is that the number of fragments will increase caused by growing FP, as some objects may not walk linearly under serious occlusion. Nonetheless, the negative influence on our overall performance can be neglected.

Among other state-of-the-art in comparison, DBN[44] and GPDBN[44] gain the second and third places in the ranking, which both apply a Bayesian filtering approach, named *dynamic bayes network* (DBN), for state prediction. The changing appearance of each target is learned online based on a random forest formulation. Because of similar improvement in appearance modeling, their MOTA score is only inferior to ours by margin, but they have better performance on MOTP, MT and ML. MCFPHD[129] utilizes PHD filter for instantaneous multi-target state estimation. The decisions on target trajectories are made offline. Likewise, LPSFM[51] and LP3D[50] make use of linear programming and social force model for data association in 3D, which are also offline methods. The works[129, 51, 50] focus on modeling the motion patterns but do not incorporate any appearance model in their formulation, which explains why their performance is inferior to MOANA and DBN-based methods. SVT[128] explores the use of spatio-temporal hyper-graph to encode 3D constraints and appearance information, however, their appearance model is based on color histogram from a single image, which can be easily affected by appearance change. AMIR3D[100] is another new method that exploits RNNs to jointly reason multiple cues for tracking, including appearance similarity. The recently observed deep learning features are kept in a feature vector, but the history beyond a temporal window is discarded absolutely, which is less reliable than our strategy based on random update. Furthermore, the deep learning features are similar among objects within the same class, which may not perform well for discriminative appearance modeling. Finally, the unsatisfactory performance of KalmanSFM[92] is also caused by the relatively simple appearance descriptor, which is a raw pixel template that is sensitive to noise. As shown in figure 4.6, several false positives are introduced by their approach.

It is also interesting to study the performance of the state-of-the-art in computation efficiency. With CPU power only, MOANA is able to achieve real-time performance with an average processing speed of 19.4 frames per second on all the test sequences. Even though

there are many cases of occlusion and grouping of targets in these sequences that require massive comparison based on the adaptive appearance models, our runtime is not seriously degraded, because our strategy of similarity measurement based on feature distance and spatial weighting is relatively efficient. On the contrary, the computation speed of DBN-based methods using random forest is much slower and far from real time. The offline methods[92, 129, 51, 50] are all relatively much faster, because they either do not use appearance model or only use simple representation for their purpose. It is impressive that MOANA can gain a comparable processing speed with them while capable of running online.

4.5.2 Comparison with the State-of-the-art in 2D

Because of the application of camera self-calibration, the provided camera matrices in the *MOTChallenge* 3D benchmark are not adopted in our 3D MOT computation, but are only considered for evaluation. Therefore, our algorithm actually only leverages 2D information for 3D MOT. Our superior performance over the state-of-the-art in 3D MOT verifies the effectiveness of our self-calibration scheme.

The two test sequences in the *MOTChallenge* 3D benchmark, *AVG-TownCentre* and *PETS09-S2L2*, are also included in the 2D benchmark. The proposed method is also compared with some of the state-of-the-art 2D MOT methods on these sequences. The experimental results are respectively presented in table 4.3 and table 4.4. Note that because a different evaluation scheme for object localization is adopted in the 2D MOT dataset, the MOTP scores of all the methods are generally higher than those in the 3D MOT benchmark. Nonetheless, our proposed algorithm still demonstrates significant advantage in MOTA and ID Sw. against them.

4.5.3 Ablation Study

We conduct more experiments with variants of our proposed method on the training sequences of the *MOTChallenge* 3D benchmark, as the test sequences are not allowed for self-comparison. The results are summarized in table 4.5.

Tracker	MOTA	MOTP	MT	ML	FP	FN	ID Sw.	Frag
MOANA	**46.1**	55.1	26.1	24.8	773	3,020	60	**200**
AP_HWDPL_p[13]	28.4	66.9	4.0	27.9	941	4,005	169	412
AMIR15[100]	36.2	69.5	26.1	**17.7**	1,448	**2,882**	234	389
JointMC[43]	43.1	69.8	**29.2**	32.3	922	3,116	**28**	213
HybridDAT[137]	29.2	69.0	9.3	43.4	**532**	4,465	61	246
AM[19]	37.5	68.1	14.2	30.5	645	3,742	79	332
TSMLCDEnew[124]	33.9	68.9	20.4	31.0	997	3,604	126	274
QuadMOT[103]	30.8	69.8	18.1	31.4	1,191	3,643	111	409
NOMT[16]	31.6	**70.1**	11.1	36.3	681	4,060	146	233
DCCRF[143]	32.3	68.9	12.4	29.2	777	3,831	229	229

Table 4.3: This is the comparison of the proposed method with the state-of-the-art on the *MOTChallenge* 2D benchmark (*AVG-TownCentre*). Bold entries indicate the best results in the corresponding columns.

The proposed adaptive appearance models are applied to two data association schemes, namely cross-matching and re-identification, respectively. As we have expected, when both schemes are taken into account, we achieve the best performance in the majority of measurements. When cross-matching is not considered, a large number of identity switches occur, because of spatial ambiguity among adjacent targets. On the other hand, when re-identification is not adopted, the identities of temporarily occluded targets cannot be recovered, which also leads to inferior performance. We also compare with the appearance model from the *raw pixel template* (RPT), *i.e.*, the latest available instance from a single frame. The main difference is that a long-term history of appearance change is learned by our proposed appearance model. As can be seen from the comparison, RPT with the proposed cross-matching and re-identification schemes fails to recover most of the identity switches. Furthermore, we also evaluate the proposed adaptive update of learning rates, *i.e.*, equation 4.8. The experimental results prove the effectiveness of adaptive learning in our formulation, as more diverse

Tracker	MOTA	MOTP	MT	ML	FP	FN	ID Sw.	Frag
MOANA	**57.6**	57.0	**40.5**	7.1	1,453	**2,531**	**107**	386
AP_HWDPL_p[13]	38.9	70.8	2.4	9.5	**552**	5,164	179	328
AMIR15[100]	47.0	70.5	11.9	9.5	616	4,236	254	397
JointMC[43]	56.0	71.4	23.8	**4.8**	942	3,162	142	220
HybridDAT[137]	47.7	69.3	11.9	9.5	616	4,236	254	349
AM[19]	47.7	69.2	16.7	14.3	718	4,206	115	356
TSMLCDEnew[124]	51.5	70.6	14.3	9.5	905	3,602	165	**198**
QuadMOT[103]	49.0	**72.6**	16.7	7.1	686	3,947	285	380
NOMT[16]	53.4	70.5	14.3	9.5	884	3,465	142	208
DCCRF[143]	45.6	72.4	9.5	9.5	664	4,335	245	245

Table 4.4: This is the comparison of the proposed method with the state-of-the-art on the *MOTChallenge* 2D benchmark (*PETS09-S2L2*). Bold entries indicate the best results in the corresponding columns.

Tracker	MOTA	MOTP	MT	ML	ID Sw.
MOANA	**81.5**	70.8	**89.7**	**0.0**	**3**
MOANA w/o cross-matching	68.1	68.7	51.7	24.1	47
MOANA w/o re-identification	64.1	69.6	62.1	10.3	34
RPT w/ cross-matching & re-id	64.3	70.0	51.7	27.6	32
MOANA w/o adaptive update	77.7	69.7	82.8	6.9	12
MOANA w/o spatial weighting	80.5	70.5	86.2	6.9	7
Baseline	77.5	**72.0**	79.3	3.4	37

Table 4.5: This is the comparison of variants of MOANA on the *MOTChallenge* 3D benchmark (training sequences). Bold entries indicate the best results in the corresponding columns.

feature values are kept in our appearance model. Then, to validate the proposed Gaussian spatial weighting scheme for pixel-based appearance modeling, *i.e.*, equation 4.9, we also compare to model update without spatial weighting. As shown in table 4.5, our proposed scheme boosts the performance, as the background area is suppressed in feature extraction. Finally, MOANA also demonstrates major improvement over the baseline, especially in the reduction of identity switches.

4.5.4 Comparison of Feature Combination

In this subsection, we explore the effectiveness and computation efficiency of different features and their combinations for the proposed appearance modeling scheme. Experiments are conducted on the training set of the *MOTChallenge* 3D benchmark. The experimental results are presented in table 4.6. Note that the CNN features are extracted from a GoogLeNet[107] pre-trained on the COCO benchmark[61], with a feature dimension of 1,024. For the histogram-based features, all the feature channels have 8 bins each. The Gaussian spatial weighting scheme is not applied to the extraction of CNN features, but it is employed for the pixel-based description and histogram construction. For all the feature comparison, we adopt the Euclidean distance.

The CNN features and the combination of all pixel-based features, *i.e.*, RGB, LBP and gradient, achieve the best overall performance on the major evaluation metrics. The deep learning features are trained to classify objects with millions of samples, so they lead to higher accuracy in data association, but the feature extraction without GPU is time-consuming. The pixel-based methods demonstrate higher accuracy compared to histogram-based ones, because the spatial feature distribution is explicitly encoded in the pixel templates. We can also learn that the RGB color component contains the richest information in appearance description, as all combinations with the RGB feature generally perform better than others. Finally, the combination of RGB and LBP in pixel templates is chosen for the experiments on the test sequences, because of its robust performance and relatively lower computation requirement. Note that because the crowd of human targets is denser in the test sequences,

Tracker	MOTA	MOTP	MT	ML	ID Sw.	Hz
Pix.: RGB + LBP + Grad.	81.7	70.5	**89.7**	**0.0**	3	28.5
Pix.: RGB + LBP	81.5	**70.8**	**89.7**	**0.0**	3	29.7
Pix.: RGB + Grad.	80.4	**70.8**	82.8	**0.0**	5	30.6
Pix.: LBP + Grad.	76.3	70.0	72.4	6.9	15	31.5
Pix.: RGB	80.2	70.5	82.8	**0.0**	3	31.8
Pix.: LBP	75.7	70.2	75.9	6.9	19	36.3
Pix.: Grad.	66.5	69.7	58.6	10.3	31	32.9
Hist.: RGB + LBP + Grad.	72.6	69.9	79.3	20.7	10	36.1
Hist.: RGB + LBP	72.5	69.2	75.9	13.8	15	37.7
Hist.: RGB + Grad.	71.0	69.3	65.5	24.1	11	38.9
Hist.: LBP + Grad.	65.2	69.4	62.1	27.6	21	40.9
Hist.: RGB	70.2	70.3	75.9	17.2	14	40.7
Hist.: LBP	63.0	69.1	58.6	31.0	23	**43.2**
Hist.: Grad.	56.5	69.2	51.7	37.9	40	41.1
CNN	**82.6**	70.3	86.2	**0.0**	**1**	1.6

Table 4.6: This is the comparison of feature combinations for MOANA on the *MOTChallenge* 3D benchmark (training sequences). Bold entries indicate the best results in the corresponding columns.

which requires more computation in cross-matching and re-identification, so the general runtime is slower than the training sequences.

Chapter 5

TWO-STEP EVOLUTIONARY POSE OPTIMIZATION FOR CAMERA AND HUMANS

In this chapter, the details of the proposed algorithm for joint human and camera pose estimation in 3D are provided. The flow diagram of the proposed framework is given in figure 5.1. We first leverage the state-of-the-art in CNN-based 2D pose estimation to detect the human joint points in each video frame. To handle false positives and false negatives, an optical-flow-based tracking scheme is employed to refine the 2D locations of joint points. Meanwhile, we can initialize the camera positions through VO, where the feature points on the human body/ies are filtered away by mask(s) dilated from 2D skeleton(s), so that the movement of human body/ies does not affect the prediction of camera motion. Then the camera pose and human joint points are jointly optimized through a two-step EDA formulation. The objective function for camera pose optimization is defined by the reprojection error of human joint points. On the other hand, the cost of 3D human pose estimation consists of constancy in space and time, body "flatness" and joint angle constraints. The two processes iterate until the stopping criterion is met. Further details about the proposed method are introduced in the following sections.

5.1 2D Human Pose Estimation and Refinement by Feature Tracking

As shown in figure 5.1, 2EPOCH is a two-stage framework for 3D pose estimation, which means the 3D human joint points are derived through regression from detected 2D joint points. For the detection of 2D joint points, we apply the state-of-the-art in human pose estimation, *OpenPose*[10], for frame-by-frame processing. *OpenPose* is a two-branch CNN architecture for multi-person pose estimation, which jointly predicts confidence maps for

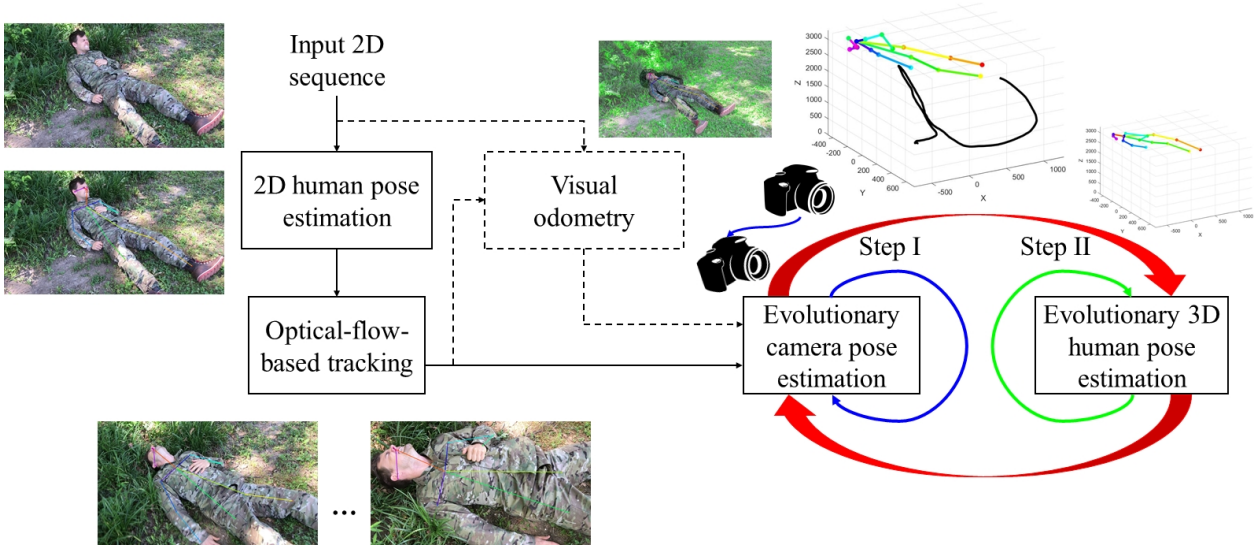

Figure 5.1: This is the flow diagram of 2EPOCH for joint human and camera pose estimation in 3D.

body part detection and part affinity fields for part association. The parsing step performs a set of bipartite matching to associate body parts candidates and finally assemble them to full body poses for multiple people in the image. The pre-trianed model for the 18-point COCO standard is adopted in this work, which is different from the 15-point MPI standard, as shown in figure 5.2. Although *OpenPose* achieves state-of-the-art performance on the MPII multi-person dataset[1] and the MSCOCO 2016 keypoints challenge[61], the false negatives and false positives in video-based pose estimation are not negligible, as the connectivity across frames is not exploited.

In our framework, the 2D poses estimated by *OpenPose* are post-processed by the *Kanade-Lucas-Tomasi* (KLT) feature tracker[68, 102], which is based on optical flow, for further refinement. The application of optical flow is based on the assumptions that the pixel intensities of an object do not change between consecutive frames and the neighboring pixels have similar motion. In our implementation, for each detected 2D joint point, we

Figure 5.2: This is the demonstration of 2D pose estimation using OpenPose[10], where the 18-point COCO standard is shown on the left, and the 15-point MPI standard on the right.

search the pixel location in the next frame within a window size of 21×21 at each pyramid level (to handle large motions). The image gradients and the gradient along time for all the pixels within the search window are computed and used to solve a set of over-determined equations with least square fit that gives the coordinates of the new pixel location. If the newly detected joint point has low confidence in *OpenPose* or the location is too far away from the prediction by tracking, it is treated as an outlier, which will be replaced by the tracking prediction.

The final output is a list of refined 2D joint points for each human object at each video frame t,

$$p(t) = \{p_{i,j}(t) : i = 1, 2, \ldots, M \quad \text{and} \quad j = 1, 2, \ldots, N\}, \tag{5.1}$$

where i indexes human objects and M is the number of humans in the current frame. And j indexes body joint points, whereas N denotes the number of joint points for each human, set as 18 in the COCO standard.

5.2 Visual Odometry and Human Motion Removal

For the optimization of camera motion, the initial camera poses can either be adopted from the previous frame, or pre-processed by visual odometry. Note that monocular VO can only estimate the relative camera movement, whose scale against real motion needs to be measured or fine-tuned from experiments. The optimization by EDA can help reduce "scale drift", *i.e.*, accumulation of small errors, in VO.

For the initialization of camera pose by VO, we first use FAST corner detector[99] to detect features in the current frame image. More specifically, a circle with 16-pixel circumference is drawn around each pixel point. For the circumference of this circle, we check if there exist a continuous set of pixels whose intensities exceed the original pixel by a certain factor, and another set of contiguous pixels with intensities less by at least the same factor. If the condition is satisfied, the pixel point is marked as a corner. FAST is selected due to its computational efficiency compared to other point detectors such as SIFT[67]. Then the KLT tracker[68, 102] is again used to search for the new feature locations, which looks around every corner point to be tracked, and uses the local information to find the corner in the next frame. Note that when the number of tracked feature points drops below a pre-defined level, the re-detection is triggered to generate more FAST corners.

Once the point-correspondences between two frames are established, the classic five-point algorithm by Nister *et al.*[87] is applied to the computation of the essential matrix. In their RANSAC formulation, five points from the set of correspondences are randomly sampled to estimate the essential matrix, and the one with the maximum number of inliers is chosen. Finally, the rotation matrix $\mathbf{R}$ and translation matrix $\mathbf{t}$ that represent the camera movement (see equation 3.11) are computed by taking the *singular-value decomposition* (SVD) of the essential matrix.

The major limitation of VO is that it must assume the camera scene is static. But in our scenario, since the motion of human objects cannot be neglected, the camera trajectory estimated from VO will be noisy. To address this problem, we propose to build a mask of

Figure 5.3: This is the demonstration of visual odometry with human motion removed by a mask constructed from the 2D skeleton. The green points are the detected and tracked corner points for VO.

each human body through morphological dilation from the joint pairs on the 2D skeleton. The kernel size of each dilation is proportional to the length of the corresponding joint pair based on empirical measurements on the human body. All the corner points detected on the human body/ies are removed from the point-correspondence set for the essential matrix estimation, so that the human motion does not affect the prediction of VO, as demonstrated in figure 5.3.

5.3 3D Human Pose Estimation by EDA

From the 2D human poses and initialized camera positions, we propose to estimate the 3D human poses through regression based on EDA. First, the cost function is designed as the

weighted sum of (1) spatial constancy loss, (2) temporal constancy loss, (3) body flatness loss, and (4) joint angle loss. Second, to reduce the dimension of multi-variate optimization for better convergence, instead of optimizing all the X, Y and Z coordinates of each joint point, we assume that the 3D joint points should lie on the optical lines connecting the corresponding 2D points and the camera center, so that only the root-relative depths, $D_{i,j}(t)$, are optimized. The number of variables can thus be reduced from $3N$ to N. Note that each person is optimized individually in the EDA process. Finally, the 3D poses will be further optimized after the camera pose is updated in the iterative two-step formulation, which will terminate until the stopping criterion is met. The details of the algorithmic design are explained as follows.

5.3.1 Spatial Constancy Loss

It is intuitive that the bone lengths of a human body should remain constant in the 3D space, which is known as *bone length constancy*[35, 123] in 3D pose estimation and body shape estimation. Like many other methods[28, 122, 123, 142], we enforce a 3D prior measured from an average human body, as shown in figure 5.4. Note that when the body length(s) of the human(s) in the scene are known, we can use their real measurements to replace the pre-defined prior. The spatial constancy loss is designed as the expected value of (fuzzy) relative bone length difference in order to reduce projection ambiguity,

$$l_{\text{spat}} = \text{E}[\max(0, \frac{|\|P_{i,j_1}(t) - P_{i,j_2}(t)\|_2 - L_{j_1,j_2}|}{L_{j_1,j_2}} - \tau_{\text{spat}})]. \tag{5.2}$$

In the COCO standard, we define 17 joint pairs in total for measurements, where j_1 and j_2 are the indices for the endpoints of each joint pair (bone length). L_{j_1,j_2} denotes the corresponding bone length in the 3D body prior, and $P_{i,j}(t)$ the estimated j'th 3D joint point for the i'th person from the sampled root-relative depth. We use τ_{spat} to represent the threshold parameter for the relative bone lengths in the fuzzy logic, empirically set as 0.1 in our experiments.

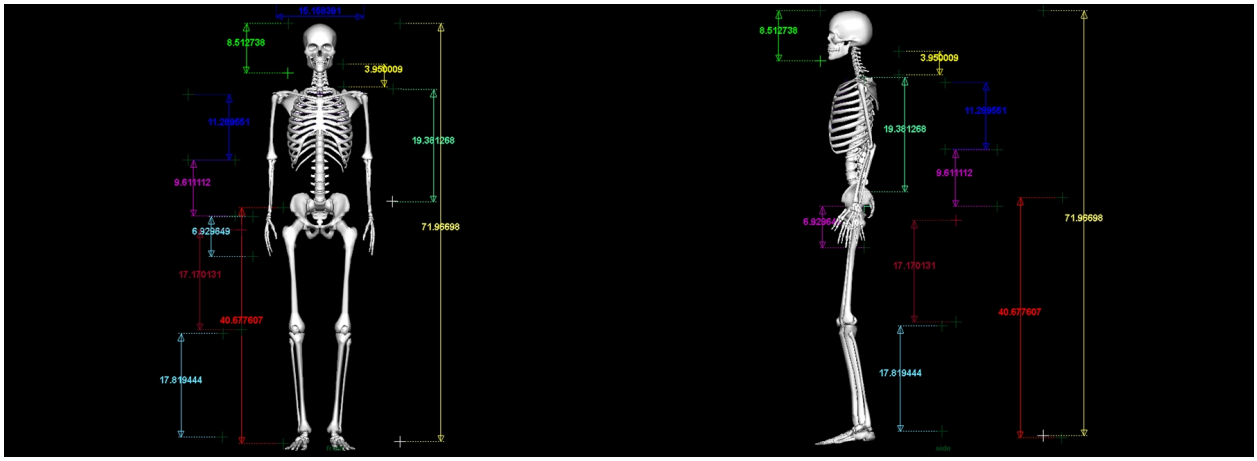

Figure 5.4: This is the 3D human body prior for the computation of spatial constancy loss. The measurements of bone lengths are in inch.

5.3.2 Temporal Constancy Loss

In a continuous video, the 3D joint points of a person should not largely deviate from their locations in the previous Δ'th frame, where Δ is usually smaller than 1 second. Hence, the temporal constancy loss is defined as the Euclidean distance between the current pose and the previous pose,

$$l_{\text{temp}} = \mathrm{E}[\|P_{i,j}(t) - P_{i,j}(t-\Delta)\|_2], \tag{5.3}$$

in which $P_{i,j}(t)$ is the 3D location of the j'th joint point (for the i'th person) at time t.

5.3.3 Body Flatness Loss

In general, the human body is considered "flat", which means if we fit a 3D plane using all the joint points, their distances to the plane should be small. To fit the plane, we first apply *principal component analysis* (PCA) on all the sampled 3D joint points of a human body. The centroid of the 3D body plane is denoted as C_i and the third eigenvector, *i.e.*, the

normal vector to the 3D plane, is represented by $\mathbf{n}$. Thus, the body flatness loss is defined as the fuzzy mean distance between joint points and their fitted body plane,

$$l_{\text{flat}} = \max(0, \mathrm{E}[|\mathbf{n} \cdot (P_{i,j}(t) - C_i)|] - \tau_{\text{flat}}), \tag{5.4}$$

in which τ_{flat} is the threshold for fuzzy logic.

5.3.4 Joint Angle Loss

The Euler angles between some joint pairs need to be limited within specified ranges, *e.g.*, the angle between the vector from the neck to the left shoulder and the vector from the neck to the right shoulder is usually be within $[110, 160]$ degrees. We design similar angular constraints for the body joint pairs $[\theta^{\min}_{j_1,j_2,j_3,j_4}, \theta^{\max}_{j_1,j_2,j_3,j_4}]$, where each $\{j_1, j_2, j_3, j_4\}$ defines the two pairs of vector endpoints that are used to measure a joint angle. The joint angle loss is thus defined as

$$\begin{aligned} l_{\text{ang}} &= \mathrm{E}[\delta_{j_1,j_2,j_3,j_4}], \\ \text{s.t.}, \delta_{j_1,j_2,j_3,j_4} &= \begin{cases} 1, & \arccos\left[\frac{\overrightarrow{P_{i,j_1}P_{i,j_2}} \cdot \overrightarrow{P_{i,j_3}P_{i,j_4}}}{\|\overrightarrow{P_{i,j_1}P_{i,j_2}}\| \cdot \|\overrightarrow{P_{i,j_3}P_{i,j_4}}\|}\right] \in [\theta^{\min}_{j_1,j_2,j_3,j_4}, \theta^{\max}_{j_1,j_2,j_3,j_4}] \\ 0, & \text{otherwise} \end{cases}. \end{aligned} \tag{5.5}$$

5.3.5 Evolutionary Optimization for 3D Joint Points

The objective function of this optimization problem combines the above loss functions,

$$\begin{aligned} P^*(t) \xleftarrow{\mathbf{P}^*(t), p(t)} D^*(t) = \underset{D(t) \in \mathrm{Rng}_{D(t)}}{\arg\min} \; & [(c + \lambda_{\text{spat}} l_{\text{spat}}) \cdot (c + \lambda_{\text{temp}} l_{\text{temp}}) \\ & \cdot (c + \lambda_{\text{flat}} l_{\text{flat}}) \cdot (c + \lambda_{\text{ang}} l_{\text{ang}})], \\ \text{s.t.,} \quad D(t) = \{D_{i,j}(t)\} \quad & \text{and} \quad D_{i,j}(t) = \|P_{i,j}(t) - \mathbf{t}^*(t)\|_2, \end{aligned} \tag{5.6}$$

where λ_{spat}, λ_{temp}, λ_{flat} and λ_{ang} are the regularization factors for the corresponding loss functions. And c is a constant (set as 1) for avoiding zero loss value(s) that dominate the

combined objective function. The camera projection matrix $\mathbf{P}^*(t) = \mathbf{K} \cdot [\mathbf{R}^*(t)|\mathbf{t}^*(t)]$ is derived from the updated camera pose $[\mathbf{R}^*(t)|\mathbf{t}^*(t)]$ ($\mathbf{K}$ is predefined), and $\mathbf{t}^*(t)$ stands for the camera location at the current frame. A 2D joint point $p_{i,j}(t)$ can be back projected to 3D ($P_{i,j}(t)$) using $\mathbf{P}^*(t)$ and the sampled root-relative depth $D_{i,j}(t)$. Our goal is to find the optimal set of 3D joint points $P^*(t)$ from the optimal root-relative depths $D^*(t)$ that lead to the lowest cost. $\text{Rng}_{D(t)}$ is the initial range for evolutionary optimization which is larger at the initial frame, and a smaller range relative to the previous frame can be used for all the following frames.

This objective can be effectively solved using EDA. We follow the convention[49] to describe the algorithmic procedure by pseudocode (see algorithm 5). Most of the configuration settings in this formulation are the same as algorithm 1.

5.4 Camera Pose Optimization by EDA

We assume that the world coordinate system aligns with the camera coordinate system at the starting frame, and thus only the 3D human poses are estimated in the two-step iteration at the beginning. In all the following frames, the camera pose can be optimized using the updated 3D human poses from the previous step. In the step of camera pose optimization, our goal is to find the optimal camera movement, equivalent to the optimal rotation and translation $[\mathbf{R}^*(t)|\mathbf{t}^*(t)]$. With the intrinsic parameter matrix $\mathbf{K}$ that is constant and assumed known, the optimal camera projection matrix in the current frame can be computed as $\mathbf{P}^*(t) = \mathbf{K} \cdot [\mathbf{R}^*(t)|\mathbf{t}^*(t)]$.

The objective function here is defined based on the reprojection errors of all the human objects' updated joint points,

$$\begin{aligned} \mathbf{P}^*(t) \xleftarrow{\mathbf{K}} [\mathbf{R}^*(t)|\mathbf{t}^*(t)] = \underset{[\mathbf{R}(t)|\mathbf{t}(t)] \in \text{Rng}_{[\mathbf{R}(t)|\mathbf{t}(t)]}}{\arg\min} & \text{E}\left[\|p_{i,j}(t) - \mathbf{P}(t) \times P^*_{i,j}(t)\|_2\right], \\ \text{s.t.,} \quad & \mathbf{P}(t) = \mathbf{K} \cdot [\mathbf{R}(t)|\mathbf{t}(t)], \end{aligned} \tag{5.7}$$

where $p_{i,j}(t)$ and $P^*_{i,j}(t)$ denote the 2D and (updated) 3D locations of the i'th person's

Algorithm 5 3D human pose estimation by EDA

Require: initial range $\mathrm{Rng}_{D(t)}$, size of initial population R, size of selected population $N < R$, maximum number of generations $g_{\max}$, stopping threshold of decreasing ratio τ_r, updated camera projection matrix $\mathbf{P}^*(t)$, estimated 2D pose of the object $p(t)$, regularization parameters λ_{spat}, λ_{temp}, λ_{flat} and λ_{ang}

Ensure: optimal 3D human pose at the current frame $P^*(t)$

1: generate initial population $\mathcal{P}(0) \leftarrow R$ sets of root-relative depths sampled uniformly in the 18D space within $\mathrm{Rng}_{D(t)}$;
2: $g \leftarrow 0$;
3: **while** ($g > 1$ and $\frac{\mu_{g-2}-\mu_{g-1}}{\mu_{g-2}} > \tau_r$) and $g < g_{\max}$ **do**
4: acquire each set of root-relative depths from $\mathcal{P}(g)$;
5: compute the 3D pose using $\mathbf{P}^*(t)$ and $p(t)$, *i.e.*, $P(t) \xleftarrow{\mathbf{P}^*(t),p(t)} D(t)$;
6: select the population of promising solutions $\mathcal{S}(g) \leftarrow N$ individuals within $\mathcal{P}(g)$ with smaller cost values $c_g = (c + \lambda_{\mathrm{spat}} l_{\mathrm{spat}}) \cdot (c + \lambda_{\mathrm{temp}} l_{\mathrm{temp}}) \cdot (c + \lambda_{\mathrm{flat}} l_{\mathrm{flat}}) \cdot (c + \lambda_{\mathrm{ang}} l_{\mathrm{ang}})$;
7: build probabilistic model $\mathcal{M}(g) = \mathcal{N}(\mu_g, \sigma_g) \leftarrow$ eighteen-variate normal density function modeled from $\mathcal{S}(g)$;
8: $\mathcal{P}(g+1) \leftarrow R$ individuals sampled from $\mathcal{M}(g)$;
9: $g \leftarrow g + 1$;
10: **end while**
11: $D^*(t) \leftarrow \mu_g$ of $\mathcal{M}(g)$;
12: output the optimal 3D pose $P^*(t) \xleftarrow{\mathbf{P}^*(t),p(t)} D^*(t)$.

j'th joint point, respectively. $\text{Rng}_{[\mathbf{R}(t)|\mathbf{t}(t)]}$ gives the initial range of rotation and translation for optimization, which is set empirically according to the video frame rate. When VO is applied, we can use smaller range of camera movement around the initialized camera pose for optimization. Again, this optimization problem can be effectively solved using EDA, where the configuration parameters are mostly set the same as algorithm 1. The pseudocode is provided in algorithm 6.

Finally, the two optimization steps, *i.e.*, 3D human pose estimation and camera pose estimation, iterate until the changing ratio of the cost value drops to certain level, or the number of iterations exceed a certain threshold.

5.5 Experimental Results

In this section, we present the qualitative experimental results on our own, benchmark and online video sequences. For quantitative evaluation, to the best of our knowledge there is no existing benchmark dataset for 3D human pose estimation captured under moving camera(s), so we experiment the proposed method on the *Human3.6M* benchmark[41, 40] that is created with a static camera array, in which our camera pose refinement step is disabled.

The characteristics of the experimental videos for qualitative evaluation are summarized in table 5.1. The visualization of 3D pose estimation on four of the video sequences, where the movement of the objects is small, is shown in figure 5.5. Seq. #1, #2 and #3 are from the same video compressed into different quality, and Seq. #4 is a low-quality video captured in similar style. As demonstrated, our algorithm can reliably recover both the 3D human poses and the camera trajectories, even when videos are highly compressed. Though the performance degrades as the compression artifacts become strong, the robustness is still acceptable for application in a low-bandwidth network. We also experiment with videos in which the objects' motion is more complex (see figure 5.6), including one from the public benchmark, DALY[127], which further verify the effectiveness of the proposed framework. In Seq. #7 and #8, we demonstrate our capability of handling multi-person pose estimation, which is applicable for either a static camera or a moving camera. The experiments are all

Algorithm 6 Optimization of camera pose by EDA

Require: initial range $\mathrm{Rng}_{[\mathbf{R}(t)|\mathbf{t}(t)]}$, size of initial population R, size of selected population $N < R$, maximum number of generations $g_{\max}$, stopping threshold of decreasing ratio τ_r, predefined intrinsic parameter matrix $\mathbf{K}$

Ensure: optimal camera projection matrix $\mathbf{P}^*(t)$

1: generate initial population $\mathcal{P}(0) \leftarrow R$ sets of extrinsic parameters γ, α, β, t_X, t_Y and t_Z sampled uniformly in the 6D space within $\mathrm{Rng}_{[\mathbf{R}(t)|\mathbf{t}(t)]}$;
2: $g \leftarrow 0$;
3: **while** ($g > 1$ and $\frac{\mu_{g-2}-\mu_{g-1}}{\mu_{g-2}} > \tau_r$) and $g < g_{\max}$ **do**
4: acquire each set of extrinsic camera parameters from $\mathcal{P}(g)$;
5: compute the extrinsic parameter matrix $[\mathbf{R}(t)|\mathbf{t}(t)]$ using equation 3.11;
6: compute the projection matrix $\mathbf{P}(t) \leftarrow \mathbf{K} \cdot [\mathbf{R}(t)|\mathbf{t}(t)]$
7: project each 3D joint point to 2D using the projection matrix $\hat{p_{i,j}}(t) \leftarrow \mathbf{P}(t) \times P_{i,j}(t)$;
8: measure each Euclidean distance between $p_{i,j}(t)$ and $\hat{p_{i,j}}(t)$;
9: select the population of promising solutions $\mathcal{S}(g) \leftarrow N$ individuals within $\mathcal{P}(g)$ that have smaller cost values $c_g = \mathrm{E}\left[\|p_{i,j}(t) - \hat{p_{i,j}}(t)\|_2\right]$;
10: build probabilistic model $\mathcal{M}(g) = \mathcal{N}(\mu_g, \sigma_g) \leftarrow$ six-variate normal density function modeled from $\mathcal{S}(g)$;
11: $\mathcal{P}(g+1) \leftarrow R$ individuals sampled from $\mathcal{M}(g)$;
12: $g \leftarrow g + 1$;
13: **end while**
14: $[\mathbf{R}^*(t)|\mathbf{t}^*(t)] \leftarrow \mu_g$ of $\mathcal{M}(g)$;
15: output the optimal camera projection matrix $P^*(t) \xleftarrow{\mathbf{K}} [\mathbf{R}^*(t)|\mathbf{t}^*(t)]$.

Seq. ID	Source	# people	Len.	Res. (pix.)	Hz (fps)	Size
1	Ours	1	17 s 667 ms	1920×1080	30	33.7 MB
2	Ours	1	17 s 667 ms	536×302	20	1.03 MB
3	Ours	1	17 s 667 ms	512×288	3	325 KB
4	Ours	1	8 s 0 ms	256×144	3	58.1 KB
5	Ours	1	12 s 375 ms	640×480	8	1.98 MB
6	DALY[127]	1	59 s 837 ms	1280×720	29.97	10.6 MB
7	Online	3	20 s 540 ms	1920×1080	29.97	3.34 MB
8	Online	3	12 s 540 ms	1280×720	30	22.5 MB

Table 5.1: This table lists the details of test sequences for qualitative evaluation of camera pose estimation and 3D human pose estimation.

conducted with an external GPU (NVIDIA Titan Xp) installed in a Razer Core V2.

The quantitative experimental results on the *Human3.6M* benchmark compared with several state-of-the-art algorithms is given in table 5.2. We show the comparison on three of the scenarios, where the evaluation metric is *Mean Per Joint Position Error* (MPJPE) in millimeter. Note that the proposed method does not rely on supervised learning, which is different from all the other listed approaches. Nonetheless, 2EPOCH still achieves performance comparable with some state-of-the-art based on deep learning. Besides, our proposed method is also capable of recovering the camera trajectories, which cannot be evaluated on this benchmark, since all the cameras are static.

Method	Walk Dog	Walk	Walk Together	Avg.
Du *et al.*[27]	137.4	99.3	106.5	114.4
Sanzari *et al.*[101]	130.5	92.6	102.2	108.4
Zhou *et al.*[144]	114.2	79.4	97.7	97.1
Nie *et al.*[85]	90.6	86.0	89.5	88.7
Tekin *et al.*[118]	126.3	55.1	65.8	82.4
Rogez *et al.*[98]	86.6	64.9	84.0	78.5
Tome *et al.*[120]	86.3	71.4	73.1	76.9
Moreno-Noguer[80]	73.5	81.6	72.6	75.9
Chen and Ramanan[12]	55.7	85.9	62.5	68.0
Tekin *et al.*[117]	74.3	51.8	74.3	66.8
Pavlakos *et al.*[91]	74.9	59.1	63.2	65.7
Mehta *et al.*[73]	55.2	76.5	61.4	64.4
Lin *et al.*[60]	50.6	72.9	57.7	60.4
Zhou *et al.*[145]	51.4	63.2	55.3	56.6
Martinez *et al.*[72]	65.1	49.5	52.4	55.7
Pavlakos *et al.*[90]	60.9	44.7	47.8	51.1
2EPOCH (proposed)	78.0	69.3	67.9	71.7

Table 5.2: This table shows the comparison with the state-of-the-art in 3D human pose estimation on the *Human3.6M* benchmark

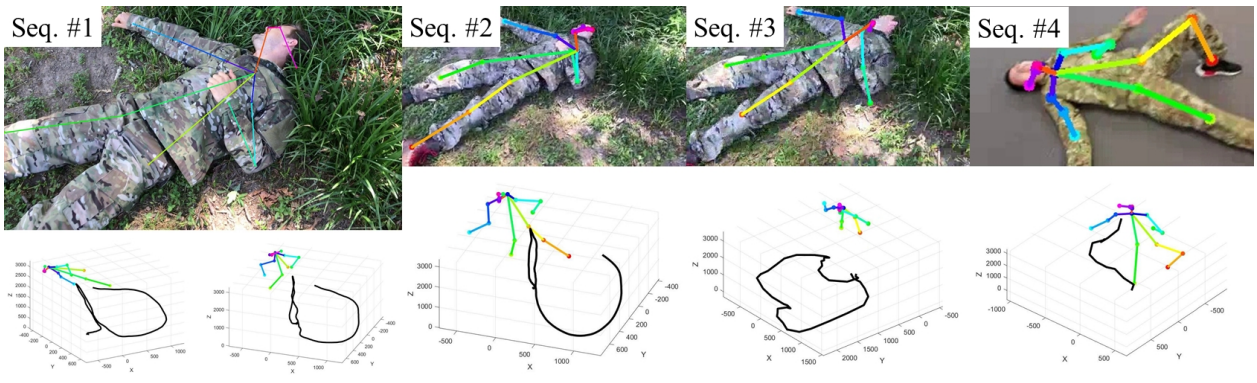

Figure 5.5: This is the visualization of 2EPOCH's qualitative performance on videos with objects whose movement is small. The unit in 3D is millimeter.

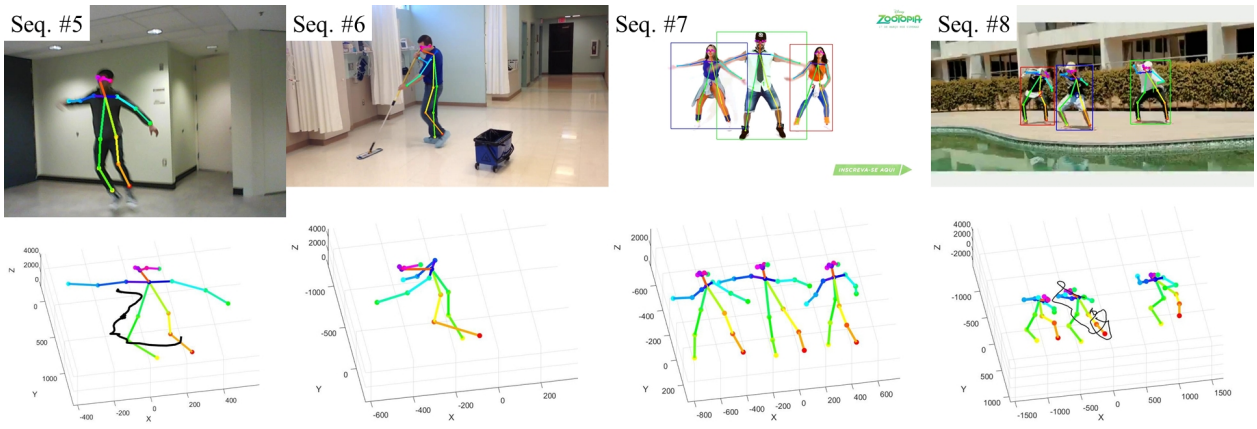

Figure 5.6: This is the qualitative visualization of the performance of 2EPOCH on videos with objects whose movement is large. The unit in 3D is millimeter.

Chapter 6

JOINT MULTI-VIEW PEOPLE TRACKING AND POSE ESTIMATION FOR 3D SCENE RECONSTRUCTION

In a single camera view, the 3D human locations and poses can be estimated from 2EPOCH. But when multiple viewing perspectives are available, we need to apply association scheme(s) and/or take advantage of the optimum view selection for each human target. Our proposed framework for multi-view scene reconstruction consists of two main steps (see figure 6.1). First, we track each target by data association across different views using visual and semantic cues, including feedback from pose estimation. Second, his/her 3D body skeleton is computed through evolutionary optimization using geo-localization information from multi-view tracking. The proposed method has been partially described in our previous publication[108].

6.1 Multi-view Object Tracking by Data Association

In each single view, we first use the state-of-the-art object detector[96] to obtain the detected bounding boxes at each frame. Then we employ a Kalman-filter-based approach[18] to associate them into tracklets. Specifically, each trajectory is fragmented either when it (1) exits from a frame border, (2) is occluded, or (3) has another trajectory that is close to it in space. All the cameras have been self-calibrated with the 2D tracklets based on ESTHER and converted to a global world coordinate system according to some shared reference point(s).

Similar to the single-camera scenario, our objective is to recover the trajectories $\mathcal{T}$ of all people within the 3D scene, that is,

$$\mathcal{T} = \{\mathcal{T}_i : i = 1, 2, \ldots, |\mathcal{T}|\}. \tag{6.1}$$

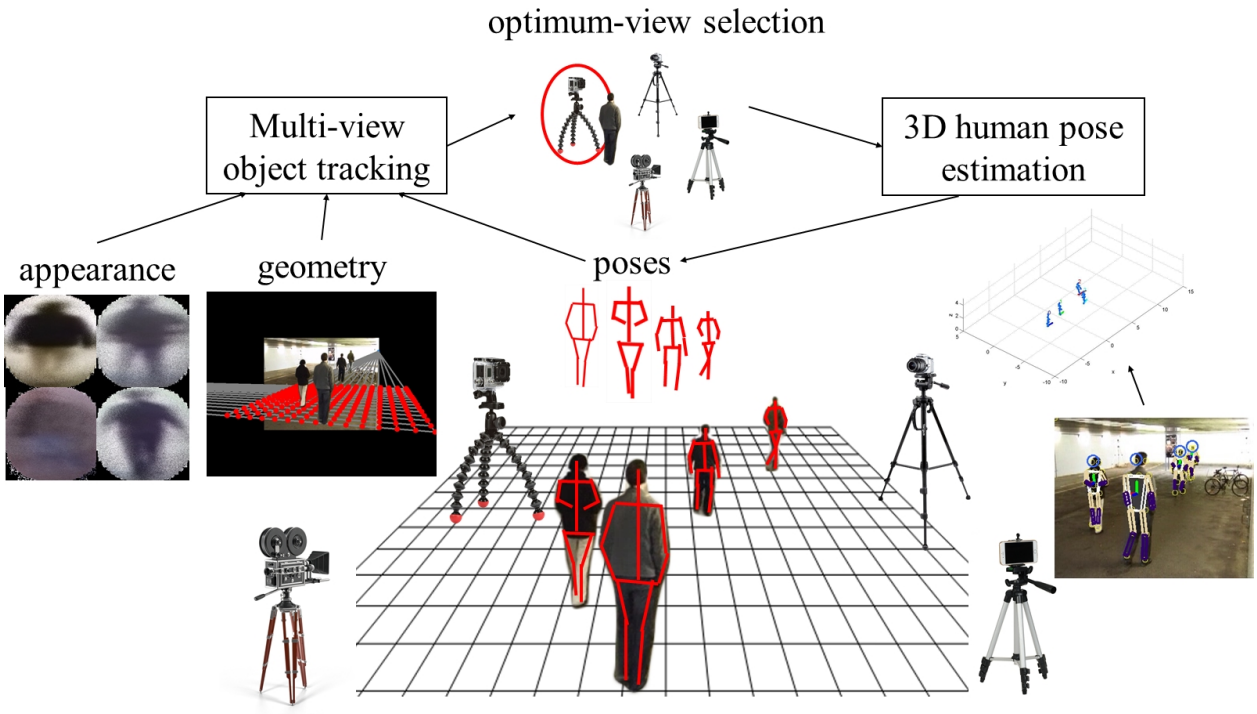

Figure 6.1: This is the illustration of the process of multi-view 3D scene reconstruction.

Tracklets T are the basic units in multi-view tracking, consisting of appearance, geometry and posture information, in a time period across multiple cameras.

$$T = \{(M_j^c, G_j^c, P_j^c) : j = 1, 2, \ldots, |T| \quad \text{and} \quad c = 1, 2, \ldots, \mathcal{C}\}. \tag{6.2}$$

where M_j^c, G_j^c and P_j^c respectively denote the appearance model, geometry information and the estimated 3D skeleton, whereas c indexes each camera and $\mathcal{C}$ is the total number of cameras. All the false positives of trajectories are collected at T_∞^c. We aim to solve the following representation:

$$W = \{\mathcal{T}_i \leftarrow T_j^c : \forall i, \forall j, \forall c\}, \tag{6.3}$$

which can be formulated as searching for the optimal solution by maximizing a posterior

$$p(W|I) \propto \exp\left[-E(W,I)\right], \tag{6.4}$$

where I denotes the input image sequences and $E(W,I)$ is the total energy function over three semantic attributes (appearance, geometry and pose):

$$E(W,I) = \sum_t (E_t^{\text{app}} + \lambda_{\text{geo}} E_t^{\text{geo}} + \lambda_{\text{pos}} E_t^{\text{pos}}), \tag{6.5}$$

in which λ_{geo} and λ_{pos} are regularization parameters. This energy minimization problem can be effectively solved by the *reversible jump Markov chain Monte Carlo* (MCMC) method[34]. The gaps between associated tracklets are interpolated linearly.

6.1.1 Appearance Attribute

The term E_t^{app} is used to describe appearance affinity of detected bounding boxes. We propose to model the appearance of each target based on MOANA. The term M_j^c is defined by a combination of $w \times h$ pixel models, which each "memorizes" a history of n observed feature values at each corresponding pixel location p. The procedure of model construction and update across multiple views is described in figure 6.2. The object region within each detected bounding box is normalized to $w \times h$ pixels masked with a maximum ellipse. Similar to single-view MOANA, the learning rates are dynamically controlled by the similarity with previously observed features and spatial weighting. In each frame, if there are less than n feature vectors at a pixel location p in M_j^c, the observed feature vector at p is added to M_j^c by a probability equivalent to the learning rate. Otherwise, a random feature vector in M_j^c is swapped by the observed feature vector with a probability equivalent to the learning rate.

Let a and b denote two different views. To compare a detected box f_k^b at the beginning of a tracklet in one view with an appearance model constructed in another view M_j^a, we adopt the color transfer method used in inter-camera tracking[55, 54] to compensate for the change of illumination and color response across different cameras. The matching score of appearance similarity is computed as

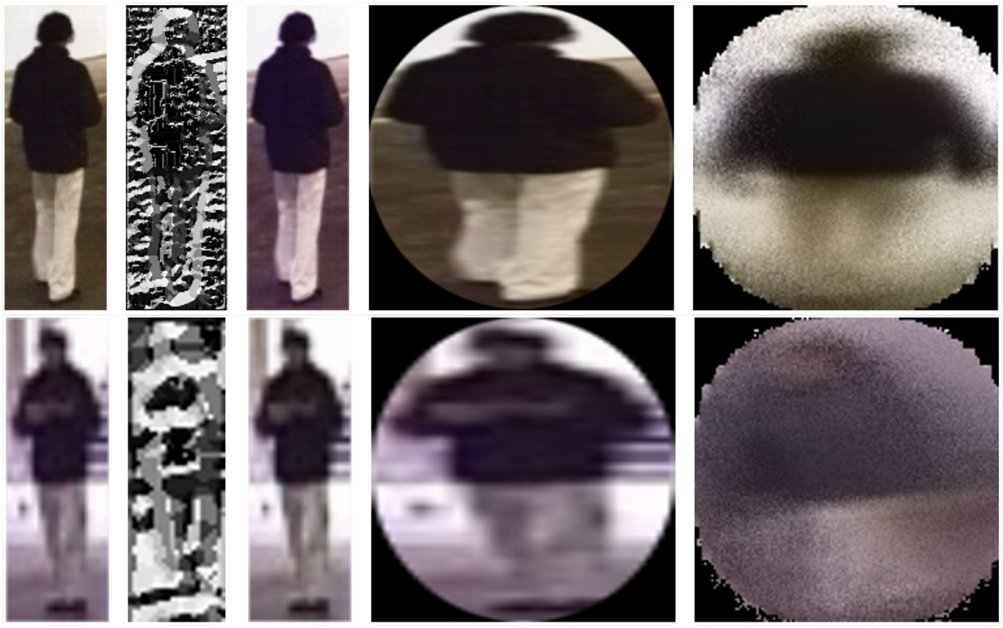

Figure 6.2: This figure demonstrates adaptive appearance models and their comparison with detected objects. The two rows give the same object identity in two different views. The columns from left to right give the RGB images, LBP images, color-transferred images, normalized bounding boxes with ellipse masks, and (averaged) appearance models (color components only).

$$s_{j,k}^{a,b} = \frac{\sum_p [\#(\|f_k^b(p) - M_{j,l}^a(p)\|_2 < \tau_f, \forall l \leq n)]}{w \cdot h \cdot n}, \tag{6.6}$$

where τ_f is the maximum feature distance threshold. The equation measures the sum of matched samples within the object region weighted by the total number of samples. Hence, we can define the objective energy for appearance affinity as

$$E_t^{\text{app}} = \text{E}\left[\frac{1}{s_{j,k}^{a,b} + s_{k,j}^{b,a}}\right], \quad \text{s.t.,} \quad \mathcal{T}_i \leftarrow T_j^a, T_k^b, \tag{6.7}$$

in which we utilize two-way comparison to enhance the robustness of the appearance de-

scriptor.

As demonstrated in figure 6.2, the RGB color values and LBP values are adopted as features for appearance modeling across multiple views. The absolute color distance threshold and LBP distance threshold are both set to 30. The dimension of each appearance model is $128 \times 128 \times 16$.

6.1.2 Geometry Attribute

The term E_t^{geo} encourages to minimize the distance of each pair of object locations assigned to the same object identity at the same frame.

Similar to equation 4.4, The geometry information of T_j^c contains four aspects.

$$G_j^c \sim (\rho_j^c, d_j^c, v_j^c, b_j^c), \tag{6.8}$$

where $\rho_j^c \in \mathbb{R}^2$ is the predicted 3D ground location in the global coordinate system, d_j^c is the depth to the camera, v_j^c is the *visibility*[52] defined as the percentage of visible area when an object is occluded by other(s), and b_j^c is an indicator of whether the bounding box is attached to a frame border. Note that b_j^c is set to 1 when the shortest distance of an edge of the bounding box to a frame border is larger than 10 pixels and 0.01 otherwise. The objective energy for geometry can be defined accordingly,

$$E_t^{\text{geo}} = \mathrm{E}\left[\|\rho_j^a - \rho_k^b\|_2 \cdot \frac{\min(v_j^a \cdot v_k^b) \cdot b_j^a \cdot b_k^b}{\max(d_j^a \cdot d_k^b)}\right], \quad \text{s.t.,} \quad \mathcal{T}_i \leftarrow T_j^a, T_k^b, \tag{6.9}$$

where $\|\rho_j^a - \rho_k^b\|_2$ computes the Euclidean distance between ρ_j^a and ρ_k^b. The 3D distance is divided by the maximum depth, because the precision of 3D localization decreases as an object moves far away from a camera. Moreover, since the estimation of foot points is prone to error when a bounding box is occluded or attached to a frame border, the objective is multiplied by the minimum visibility and the indicators of attachment to frame borders.

6.1.3 Pose Attribute

The feedback of 3D human joint points, noted $P_j^c \in \mathbb{R}^{18\times 3}$ (following the 18-point COCO standard[61]), from pose estimation is applied to the objective energy as the pose/action attribute. Different from the previous work[134], our pose descriptor is a set of joint points encoding the 3D actions explicitly. This can help avoid ambiguity in pose transitions and reduce complexity with temporal information. The pose objective is weighted by some geometry aspects similar to equation 6.9, because the estimation of human skeleton also relies on high resolution and high visibility. More specifically, E_t^{pos} is defined as follows,

$$E_t^{\text{pos}} = \mathrm{E}\left[\|P_j^a - P_k^b\|_2 \cdot \frac{\min(v_j^a \cdot v_k^b) \cdot b_j^a \cdot b_k^b}{\max(d_j^a \cdot d_k^b)}\right], \quad \text{s.t.,} \quad \mathcal{T}_i \leftarrow T_j^a, T_k^b, \tag{6.10}$$

where $\|P_j^a - P_k^b\|_2$ measures the Euclidean distance between the set of joint points shared in both views.

6.2 Evolutionary 3D Pose Estimation from Optimum View

From the geometry information in multi-view tracking, we can define the optimum view of each $\mathcal{T}_i$ at time t as

$$c_t^* = \arg\max_{\forall c \leq C} \frac{v_t^c \cdot b_t^c}{d_t^c}, \tag{6.11}$$

which is chosen for single-view 3D pose estimation. Then we follow the pipeline of 2EPOCH to estimate each 3D human pose based on the selected optimum view. Note that the step of camera pose estimation/optimization can be discarded as each camera is assumed static. The computed 3D poses are fed back to multi-view tracking as the pose attribute for data association. Finally, the estimated joint points are augmented onto the 3D trajectories for scene reconstruction.

Method	MODA (%)	MODP (%)	MOTA (%)	MOTP (%)
Proposed	**61.04**	**73.13**	**60.26**	**72.26**
HTC[133]	43.75	67.11	43.75	67.11
KSP[4]	40.46	58.88	40.46	57.24
POM[29]	32.57	62.50	32.57	60.86

Table 6.1: This is the quantitative comparison of multi-view object tracking on the EPFL benchmark. Bold entries indicate the best results in the corresponding columns.

6.3 Experimental Results

Our proposed method is evaluated on the EPFL benchmark[29]. We adopt the *passageway* sequence in our experiments, which is known for its challenging scenario with poor lighting and image quality. People can become very small on the far end and some of them are captured in only one or two cameras. The sequence consists of 4 different views and films 11 pedestrians walking or bicycling. Each view is shot at 25 fps and in a relatively low resolution 360×288.

The quantitative comparison of the proposed method with several state-of-the-art algorithms in multi-view object tracking is presented in table 6.1. The widely used CLEAR metrics[5] are adopted, including *multiple object detection accuracy* (MODA), *multiple object detection precision* (MODP), MOTA and MOTP. MOTA measures three sources of errors in detection and tracking respectively: false positives, false negatives and identity switches. On the other hand, MODA only measures the former two error sources in object detection. MODP and MOTP are used to measure misalignment between annotated and predicted locations.

The proposed algorithm achieves the top performance on all evaluation metrics in this challenging sequence. Our promising performance in tracking is mainly due to the effective formulation of multi-view object tracking by integrating robust visual and semantic attributes

including appearance, geometry and human poses. The performance HTC[133] is inferior to us by margin, as they only consider appearance and motion patterns in their hierarchical feature model. Moreover, their feature descriptor for appearance modeling is only extracted in several frames, however, the proposed adaptive appearance model can "memorize" a long history of past feature values. Both KSP[4] and POM[29] rely on a less robust appearance representation and suffer from an object localization strategy of poor accuracy (tracking by segmentation). The relatively higher MODA and MODP gained by our method and HTC[133] confirm that the tracking-by-detection-based approaches are superior in terms of localization of object observations. A qualitative demonstration of the proposed framework for multi-view scene reconstruction can be seen in figure 1.1.

Chapter 7

CONCLUSIONS

In this dissertation, we propose a novel framework for 3D scene reconstruction that can either be applied to a single camera view or multiple viewpoints. The main components include camera self-calibration, multi-target tracking in 3D, 3D pose estimation, and their extension to multi-view object tracking.

For camera self-calibration and radial distortion correction from walking humans, there are three critical challenges to overcome, *i.e.*, the relaxation of assumptions on unknown intrinsic camera parameters, the estimation of vanishing points against noise, and the automatic computation of distortion coefficients. To address these problems, we propose several innovative schemes in the process of estimation and optimization. The main contributions in this part of work in terms of novelty include: (1) evolutionary optimization of distortion coefficients based on human height variance minimization; (2) camera parameters optimization using EDA that aims to minimize the reprojection error on the ground plane; (3) mean shift clustering for the removal of outliers in the estimation of the vertical vanishing point; (4) the estimation of horizon line based on Laplace linear regression that avoids additional fine-tuning; (5) a robust segmentation and tracking system that can adaptively refine the foreground masks to support optimal head/foot localization; (6) state-of-the-art performance demonstrated on several public benchmarks and our own dataset, enabling application in 3D object tracking.

Multi-target tracking in 3D has also been a challenging task, especially because of identity switches caused by occlusion, spatial ambiguity and similar appearance among nearby targets. We propose an adaptive appearance modeling scheme to support robust MOT. Different from previous works in the development of discriminative appearance features, our

extracted feature vectors are saved in an explicit form and adaptively updated online. The proposed method is robust against appearance change due to different illumination, poses and viewing perspectives. Based on the adaptive appearance model, we design cross-matching and re-identification schemes to mitigate identity switch when objects interact with others. Besides, 3D geometry information is effectively incorporated into our formulation of data association. Experimental results on the *MOTChallenge* benchmark datasets show our superior performance in robustness and efficiency compared with the state-of-the-art.

Besides, we also develop a two-step evolutionary optimization approach to jointly estimate the 3D poses of the camera and humans. At every frame, the 2D joint points of humans are first detected through a CNN and refined by feature tracking. We can also estimate the preliminary camera pose based on visual odometry, where the feature points extracted on the human bodies are removed by masks created by dilation from the 2D skeletons, so that their motion can be ignored in the prediction of camera trajectory. Then we iteratively optimize the 3D human poses and the camera pose based on EDA. At the step of human pose estimation, we only optimize the root-relative depth of each joint point, and the cost computation combines the information of spatial and temporal constancy, body flatness and joint angle constraints. As for the camera pose optimization, the evolutionary optimization is based on the minimization of reprojection error of the joint points. The effectiveness of the proposed method is demonstrated on benchmark and in-the-wide videos of various scenarios and quality.

Finally, we also present a multi-view scene reconstruction framework jointly combining the efforts of multi-view MOT and 3D pose estimation. In multi-view people tracking, we associate targets based on rich visual and semantic attributes, including adaptive appearance models, spatially-weighted geometry measurements, and the feedback of 3D joint points from pose estimation. The data association across different views is formulated as an energy minimization problem that is solved by an MCMC-based approach. The 3D pose estimation is conducted based on the mentioned evolutionary optimization scheme, which is benefited from the optimum views selected from multi-view tracking. Experiments on a public benchmark

demonstrates the efficacy of the proposed method.

In the future, the proposed framework can be extended to camera arrays with non-overlapping views which require re-identification and spatio-temporal association. Moreover, as vehicles are also common objects in video analytics, we can also apply the proposed algorithms to 3D car tracking and shape modeling. Nonetheless, different from humans, vehicles demonstrate higher intra-class variability due to dependence of shapes on viewpoints, and higher inter-class similarity caused by similar vehicle models produced by various manufacturers. Therefore, vehicle-based scene reconstruction will requires major improvement over the existing architecture.

BIBLIOGRAPHY

[1] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2D human pose estimation: New benchmark and state of the art analysis. In *Proc. CVPR*, pages 3686–3693, 2014.

[2] Anton Andriyenko, Konrad Schindler, and Stefan Roth. Discrete-continuous optimization for multi-target tracking. In *Proc. CVPR*, pages 1926–1933, 2012.

[3] Olivier Barnich and Marc Van Droogenbroeck. ViBe: A universal background subtraction algorithm for video sequences. *IEEE TIP*, 20(6):1709–1724, 2010.

[4] Jerome Berclaz, Francois Fleuret, Engin Turetken, and Pascal Fua. Multiple object tracking using k-shortest paths optimization. *IEEE TPAMI*, 33(9):1806–1819, 2011.

[5] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: The CLEAR MOT metrics. *JIVP*, 2008.

[6] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J. Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *Proc. ECCV*, pages 561–578, 2016.

[7] Michael D. Breitenstein, Fabian Reichlin, Bastian Leibe, Esther Koller-Meier, and Luc Van Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *IEEE TPAMI*, 33(9):1820–1833, 2010.

[8] Jose Henrique Brito, Roland Angst, Kevin Koser, and Marc Pollefeys. Radial distortion self-calibration. In *Proc. CVPR*, pages 1368–1375, 2013.

[9] Guido MYE Brouwers, Matthijs H. Zwemer, and Rob GJ Wijnhoven. Automatic calibration of stationary surveillance cameras in the wild. In *Proc. ECCV*, pages 743–759, 2016.

[10] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2D pose estimation using part affinity fields. In *Proc. CVPR*, pages 7291–7299, 2017.

[11] Bruno Caprile and Vincent Torre. Using vanishing points for camera calibration. *IJCV*, 4(2):127–139, 1990.

[12] Ching-Hang Chen and Deva Ramanan. 3D human pose estimation = 2D pose estimation + matching. In *Proc. CVPR*, pages 7035–7043, 2017.

[13] Long Chen, Haizhou Ai, Chong Shang, Zijie Zhuang, and Bo Bai. Online multi-object tracking with convolutional neural networks. In *Proc. ICIP*, pages 645–649, 2017.

[14] Tsuhan Chen, Alberto Del Bimbo, Federico Pernici, and Giuseppe Serra. Accurate self-calibration of two cameras by observations of a moving person on a ground plane. In *Proc. AVSS*, pages 129–134, 2007.

[15] Xianjie Chen and Alan L. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *Proc. NeurIPS*, pages 1736–1744, 2014.

[16] Wongun Choi. Near-online multi-target tracking with aggregated local flow descriptor. In *Proc. ICCV*, pages 3029–3037, 2015.

[17] Chun-Te Chu, Jenq-Neng Hwang, Hung-I Pai, and Kung-Ming Lan. Tracking human under occlusion based on adaptive multiple kernels with projected gradients. *IEEE TMM*, 15(7):1602–1615, 2013.

[18] Chun-Te Chu, Jenq-Neng Hwang, Shen-Zheng Wang, and Yi-Yuan Chen. Human tracking by adaptive Kalman filtering and multiple kernels tracking with projected gradients. In *Proc. ICDSC*, 2011.

[19] Qi Chu, Wanli Ouyang, Hongsheng Li, Xiaogang Wang, Bin Liu, and Nenghai Yu. Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism. In *Proc. ICCV*, pages 4836–4845, 2017.

[20] Xiao Chu, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Structured feature learning for pose estimation. In *Proc. CVPR*, pages 4715–4723, 2016.

[21] Xiao Chu, Wei Yang, Wanli Ouyang, Cheng Ma, Alan L. Yuille, and Xiaogang Wang. Multi-context attention for human pose estimation. In *Proc. CVPR*, pages 1831–1840, 2017.

[22] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE TPAMI*, 24(5):603–619, 2002.

[23] Peter Corke, Dennis Strelow, and Sanjiv Singh. Omnidirectional visual odometry for a planetary rover. In *Proc. IROS*, pages 4007–4012, 2004.

[24] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, pages 886–893, 2005.

[25] Jonathan Deutscher, Michael Isard, and John MacCormick. Automatic camera calibration from a single Manhattan image. In *Proc. ECCV*, pages 175–188, 2002.

[26] Frédéric Devernay and Olivier D. Faugeras. Automatic calibration and removal of distortion from scenes of structured environments. In *Proc. ITIP*, pages 62–73, 1995.

[27] Yu Du, Yongkang Wong, Yonghao Liu, Feilin Han, Yilin Gui, Zhen Wang, Mohan Kankanhalli, and Weidong Geng. Marker-less 3D human motion capture with monocular image sequence and height-maps. In *Proc. ECCV*, pages 20–36, 2016.

[28] Ahmed Elgammal and Chan-Su Lee. Inferring 3D body pose from silhouettes using activity manifold learning. In *Proc. CVPR*, 2004.

[29] Francois Fleuret, Jerome Berclaz, Richard Lengagne, and Pascal Fua. Multicamera people tracking with a probabilistic occupancy map. *IEEE TPAMI*, 30(2):267–282, 2007.

[30] Gustavo Führ and Cláudio Rosito Jung. Combining patch matching and detection for robust pedestrian tracking in monocular calibrated cameras. *Pattern Recognition Letters*, 39:11–20, 2014.

[31] Gustavo Führ and Cláudio Rosito Jung. Camera self-calibration based on nonlinear optimization and applications in surveillance systems. *IEEE TCSVT*, 27(5):1132–1142, 2015.

[32] Nil Goyette, Pierre-Marc Jodoin, Fatih Porikli, Janusz Konrad, and Prakash Ishwar. changedetection.net: A new change detection benchmark dataset. In *Proc. CVPR Workshops*, 2012.

[33] Michael Grant and Stephen Boyd. CVX: MATLAB software for disciplined convex programming. http://cvxr.com/cvx, 2014.

[34] Peter J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

[35] Nils Hasler, Hanno Ackermann, Bodo Rosenhahn, Thorsten Thormählen, , and Hans-Peter Seidel. Multilinear pose and body shape estimation of dressed subjects from image sets. In *Proc. CVPR*, pages 1823–1830, 2010.

[36] Mark Hauschild and Martin Pelikan. An introduction and survey of estimation of distribution algorithms. *Swarm and Evolutionary Computation*, 1(3):111–128, 2011.

[37] Martin Hofmann, Philipp Tiefenbacher, and Gerhard Rigoll. Background segmentation with feedback: The pixel-based adaptive segmenter. In *Proc. CVPR Workshops*, pages 38–43, 2012.

[38] Shiyao Huang, Xianghua Ying, Jiangpeng Rong, Zeyu Shang, and Hongbin Zha. Camera calibration from periodic motion of a pedestrian. In *Proc. CVPR*, pages 3025–3033, 2016.

[39] Catalin Ionescu, Joao Carreira, and Cristian Sminchisescu. Iterated second-order label sensitive pooling for 3D human pose estimation. In *Proc. CVPR*, pages 1661–1668, 2014.

[40] Catalin Ionescu, Fuxin Li, and Cristian Sminchisescu. Latent structured models for human pose estimation. In *Proc. ICCV*, pages 2220–2227, 2011.

[41] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *IEEE TPAMI*, 36(7):1325–1339, 2013.

[42] Imran Junejo and Hassan Foroosh. Robust auto-calibration from pedestrians. In *Proc. AVSS*, 2006.

[43] Margret Keuper, Siyu Tang, Yu Zhongjie, Bjoern Andres, Thomas Brox, and Bernt Schiele. A multi-cut formulation for joint segmentation and tracking of multiple objects. arXiv:1607.06317, 2016.

[44] Tobias Klinger, Franz Rottensteiner, and Christian Heipke. Probabilistic multi-person tracking using dynamic bayes networks. In *Proc. ISPRS Annals*, pages 435–442, 2015.

[45] Tobias Klinger, Franz Rottensteiner, and Christian Heipke. Probabilistic multi-person localisation and tracking in image sequences. *ISPRS J. P&RS*, 127:73–88, 2017.

[46] Nils Krahnstoever and Paulo RS Mendona. Autocalibration from tracks of walking people. In *Proc. BMVC*, 2006.

[47] Cheng-Hao Kuo, Chang Huang, and Ramakant Nevatia. Multi-target tracking by online learned discriminative appearance models. In *Proc. CVPR*, pages 685–692, 2010.

[48] Worapan Kusakunniran, Hongdong Li, and Jian Zhang. A direct method to self-calibrate a surveillance camera by observing a walking pedestrian. In *Proc. DICTA*, pages 250–255, 2000.

[49] Pedro Larraanaga and Jose A. Lozano. *Estimation of distribution algorithms: A new tool for evolutionary computation*. Springer US, 2002.

[50] Laura Leal-Taixé, Anton Milan, Ian Reid, Stefan Roth, and Konrad Schindler. MOTChallenge 2015: Towards a benchmark for multi-target tracking. arXiv:1504.01942, 2015.

[51] Laura Leal-Taixé, Gerard Pons-Moll, and Bodo Rosenhahn. Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. In *Proc. ICCV Workshops*, pages 120–127, 2011.

[52] Kuan-Hui Lee and Jenq-Neng Hwang. On-road pedestrian tracking across multiple driving recorders. *IEEE TMM*, 17(9):1429–1438, 2015.

[53] Kuan-Hui Lee, Jenq-Neng Hwang, and Shih-I Chen. Model-based vehicle localization based on three-dimensional constrained multiple-kernel tracking. *IEEE TCSVT*, 25(1):38–50, 2014.

[54] Young-Gun Lee, Zheng Tang, and Jenq-Neng Hwang. Online-learning-based human tracking across non-overlapping cameras. *IEEE TCSVT*, 28(10):2870–2883, 2018.

[55] Young-Gun Lee, Zheng Tang, Jenq-Neng Hwang, and Zhijun Fang. Inter-camera tracking based on fully unsupervised online learning. In *Proc. ICIP*, pages 2607–2611, 2017.

[56] Maxime Lhuillier. Automatic structure and motion using a catadioptric camera. In *Proc. OMNIVIS*, 2005.

[57] Sijin Li and Antoni B. Chan. 3D human pose estimation from monocular images with deep convolutional neural network. In *Proc. ACCV*, pages 332–347, 2014.

[58] David Liebowitz, Antonio Criminisi, and Andrew Zisserman. Creating architectural models from images. *CGF*, 18(3):39–50, 1999.

[59] Horng-Horng Lin, Jen-Hui Chuang, and Tyng-Luh Liu. Regularized background adaptation: a novel learning rate control scheme for gaussian mixture modeling. *IEEE TIP*, 20(3):822–836, 2010.

[60] Mude Lin, Liang Lin, Xiaodan Liang, Keze Wang, and Hui Cheng. Recurrent 3D pose sequence machines. In *Proc. CVPR*, pages 810–819, 2017.

[61] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proc. ECCV*, pages 740–755, 2014.

[62] Yen-Shuo Lin, Kuo-Hua Lo, Hua-Tsung Chen, and Jen-Hui Chuang. Vanishing point-based image transforms for enhancement of probabilistic occupancy map-based people localization. *IEEE TIP*, 23(12):5586–5598, 2014.

[63] Jingchen Liu, Robert T. Collins, and Yanxi Liu. Surveillance camera autocalibration based on pedestrian height distributions. In *Proc. BMVC*, 2011.

[64] Jingchen Liu, Robert T. Collins, and Yanxi Liu. Robust autocalibration for a surveillance camera network. In *Proc. WACV*, pages 433–440, 2013.

[65] Tao Liu, Yong Liu, Zheng Tang, and Jenq-Neng Hwang. Adaptive ground plane estimation for moving camera-based 3D object tracking. In *Proc. MMSP*, 2017.

[66] Xiaobai Liu. Multi-view 3D human tracking in crowded scenes. In *Proc. AAAI*, pages 3553–3559, 2016.

[67] David G. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, pages 1150–1157, 1999.

[68] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. IJCAI*, pages 674–679, 1981.

[69] Wenhan Luo, Junliang Xing, Anton Milan, Xiaoqin Zhang, Wei Liu, Xiaowei Zhao, and Tae-Kyun Kim. Multiple object tracking: A literature review. arXiv: 1409.7618, 2014.

[70] Fengjun Lv, Tao Zhao, and Ramakant Nevatia. Self-calibration of a camera from video of a walking human. In *Proc. ICPR*, pages 562–567, 2002.

[71] Fengjun Lv, Tao Zhao, and Ramakant Nevatia. Camera calibration from video of a walking human. *IEEE TPAMI*, 28(9):1513–1518, 2016.

[72] Julieta Martinez, Rayat Hossain, Javier Romero, and James J. Little. A simple yet effective baseline for 3D human pose estimation. In *Proc. ICCV*, pages 2640–2649, 2017.

[73] Dushyant Mehta, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt. Monocular 3D human pose estimation in the wild using improved CNN supervision. In *Proc. 3DV*, pages 506–516, 2017.

[74] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. VNect: Real-time 3D human pose estimation with a single RGB camera. *ACM TOG*, 36(4), 2017.

[75] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. MOT16: A benchmark for multi-object tracking. arXiv:1603.00831, 2016.

[76] Anton Milan, Laura Leal-Taixé, Konrad Schindler, and Ian Reid. Joint tracking and segmentation of multiple targets. In *Proc. CVPR*, pages 5397–5406, 2015.

[77] Anton Milan, S. Hamid Rezatofighi, Anthony Dick, Ian Reid, and Konrad Schindler. Online multi-target tracking using recurrent neural networks. In *Proc. AAAI*, pages 4225–4232, 2017.

[78] Anton Milan, Stefan Roth, and Konrad Schindler. Continuous energy minimization for multitarget tracking. *IEEE TPAMI*, 36(1):58–72, 2013.

[79] Raúl Mohedano and Narciso Garcâa. Capabilities and limitations of mono-camera pedestrian-based autocalibration. In *Proc. ICIP*, pages 4705–4708, 2010.

[80] Francesc Moreno-Noguer. 3D human pose estimation from a single image via distance matrix regression. In *Proc. CVPR*, pages 2823–2832, 2017.

[81] Etienne Mouragnon, Maxime Lhuillier, Michel Dhome, Fabien Dekeyser, and Patrick Sayd. Real time localization and 3D reconstruction. In *Proc. CVPR*, pages 363–370, 2006.

[82] Kevin P. Murphy. *Machine learning: A probabilistic perspective*. The MIT Press, 2012.

[83] Milind Naphade, Zheng Tang, Ming-Ching Chang, David C. Anastasiu, Anuj Sharma, Rama Chellappa, Shuo Wang, Pranamesh Chakraborty, Tingting Huang, Jenq-Neng Hwang, and Siwei Lyu. The 2019 AI City Challenge. In *Proc. CVPR Workshops*, pages 452–460, 2019.

[84] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *Proc. ECCV*, pages 483–499, 2016.

[85] Bruce Xiaohan Nie, Ping Wei, and Song-Chun Zhu. Monocular 3D human pose estimation by predicting depth on joints. In *Proc. ICCV*, pages 3467–3475, 2017.

[86] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE TPAMI*, 26(6):756–770, 2004.

[87] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry. In *Proc. CVPR*, pages 1–8, 2004.

[88] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry for ground vehicle applications. *J. Field Robotics*, 23(1):3–20, 2006.

[89] Shaul Oron, Aharon Bar-Hillel, and Shai Avidan. Real-time tracking-with-detection for coping with viewpoint change. *MVA*, 26(4):507–518, 2015.

[90] Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Ordinal depth supervision for 3D human pose estimation. In *Proc. CVPR*, pages 7307–7316, 2018.

[91] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G. Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3D human pose. In *Proc. CVPR*, pages 7025–7034, 2017.

[92] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *Proc. ICCV*, pages 261–268, 2009.

[93] Hamed Pirsiavash, Deva Ramanan, and Charless C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *Proc. CVPR*, pages 1201–1208, 2011.

[94] Leonid Pishchulin, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. Poselet conditioned pictorial structures. In *Proc. CVPR*, pages 588–595, 2013.

[95] Horst Possegger, Matthias Rüther, Sabine SternigvThomas Mauthner, Manfred Klopschitz, Peter M. Roth, and Horst Bischof. Unsupervised calibration of camera networks and virtual PTZ cameras. In *Proc. CVWW*, 2012.

[96] Joseph Redmon and Ali Farhadi. YOLOv3: An incremental improvement. arXiv:1804.02767, 2018.

[97] Xiaofeng Ren, Alexander C. Berg, and Jitendra Malik. Recovering human body configurations using pairwise constraints between parts. In *Proc. ICCV*, pages 824–831, 2005.

[98] Gregory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. LCR-Net: Localization-classification-regression for human pose. In *Proc. CVPR*, pages 3433–3441, 2017.

[99] Edward Rosten and Tom Drummond. Fusing points and lines for high performance tracking. In *Proc. ICCV*, pages 1508–1515, 2005.

[100] Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In *Proc. ICCV*, pages 300–311, 2017.

[101] Marta Sanzari, Valsamis Ntouskos, and Fiora Pirri. Bayesian image based 3D pose estimation. In *Proc. ECCV*, pages 566–582, 2016.

[102] Jianbo Shi and Carlo Tomasi. Good features to track. In *Proc. CVPR*, pages 593–600, 1994.

[103] Jeany Son, Mooyeol Baek, Minsu Cho, and Bohyung Han. Multi-object tracking with quadruplet convolutional neural networks. In *Proc. CVPR*, pages 5620–5629, 2017.

[104] Pierre-Luc St-Charles and Guillaume-Alexandre Bilodeau. Improving background subtraction using local binary similarity patterns. In *Proc. WACV*, pages 509–515, 2014.

[105] Pierre-Luc St-Charles, Guillaume-Alexandre Bilodeau, and Robert Bergevin. Flexible background subtraction with self-balanced local sensitivity. In *Proc. CVPR Workshops*, pages 408–413, 2014.

[106] Pierre-Luc St-Charles, Guillaume-Alexandre Bilodeau, and Robert Bergevin. SuBSENSE: A universal change detection method with local adaptive sensitivity. *IEEE TIP*, 24(1):359–373, 2015.

[107] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proc. CVPR*, pages 1–9, 2015.

[108] Zheng Tang, Renshu Gu, and Jenq-Neng Hwang. Joint multi-view people tracking and pose estimation for 3D scene reconstruction. In *Proc. ICME*, 2018.

[109] Zheng Tang and Jenq-Neng Hwang. MOANA: An online learned adaptive appearance model for robust multiple object tracking in 3D. *IEEE Access*, 7(1):31934–31945, 2019.

[110] Zheng Tang, Jenq-Neng Hwang, Yen-Shuo Lin, and Jen-Hui Chuang. Multiple-kernel adaptive segmentation and tracking (MAST) for robust object tracking. In *Proc. ICASSP*, pages 1115–1119, 2016.

[111] Zheng Tang, Yen-Shuo Lin, Kuan-Hui Lee, and Jenq-Neng Hwang. ESTHER: Joint camera self-calibration and automatic radial distortion correction from tracking of walking humans. *IEEE Access*, 7(1):10754–10766, 2019.

[112] Zheng Tang, Yen-Shuo Lin, Kuan-Hui Lee, Jenq-Neng Hwang, Jen-Hui Chuang, and Zhijun Fang. Camera self-calibration from tracking of moving persons. In *Proc. ICPR*, pages 260–265, 2016.

[113] Zheng Tang, Milind Naphade, Ming-Yu Liu, Xiaodong Yang, Stan Birchfield, Shuo Wang, Ratnesh Kumar, David Anastasiu, and Jenq-Neng Hwang. CityFlow: A city-scale benchmark for multi-target multi-camera vehicle tracking and re-identification. In *Proc. CVPR*, pages 8797–8806, 2019.

[114] Zheng Tang, Gaoang Wang, Tao Liu, Young-Gun Lee, Adwin Jahn, Xu Liu, Xiaodong He, and Jenq-Neng Hwang. Multiple-kernel based vehicle tracking using 3D deformable model and camera self-calibration. arXiv:1708.06831, 2017.

[115] Zheng Tang, Gaoang Wang, Hao Xiao, Aotian Zheng, and Jenq-Neng Hwang. Single-camera and inter-camera vehicle tracking and 3D speed estimation based on fusion of visual and semantic features. In *Proc. CVPR Workshops*, pages 108–115, 2018.

[116] Bugra Tekin, Isinsu Katircioglu, Mathieu Salzmann, Vincent Lepetit, and Pascal Fua. Structured prediction of 3D human pose with deep neural networks. arXiv:1605.05180, 2016.

[117] Bugra Tekin, Pablo Márquez-Neila, Mathieu Salzmann, and Pascal Fua. Learning to fuse 2D and 3D image cues for monocular body pose estimation. In *Proc. ICCV*, pages 3941–3950, 2017.

[118] Bugra Tekin, Artem Rozantsev, Vincent Lepetit, and Pascal Fua. Direct prediction of 3D body poses from motion compensated sequences. In *Proc. CVPR*, pages 991–1000, 2016.

[119] Tai-Peng Tian and Stan Sclaroff. Fast globally optimal 2D human detection with loopy graph models. In *Proc. CVPR*, pages 81–88, 2010.

[120] Denis Tome, Chris Russell, and Lourdes Agapito. Lifting from the deep: Convolutional 3D pose estimation from a single image. In *Proc. CVPR*, pages 2500–2509, 2017.

[121] Alexander Toshev and Christian Szegedy. DeepPose: Human pose estimation via deep neural networks. In *Proc. CVPR*, pages 1653–1660, 2014.

[122] Raquel Urtasun, David J. Fleet, and Pascal Fua. 3D people tracking with gaussian process dynamical models. In *Proc. CVPR*, pages 238–245, 2006.

[123] Bastian Wandt, Hanno Ackermann, and Bodo Rosenhahn. 3D reconstruction of human motion from monocular image sequences. *IEEE TPAMI*, 38(8):1505–1516, 2016.

[124] Bing Wang, Gang Wang, Kap Luk Chan, and Li Wang. Tracklet association by online target-specific metric learning and coherent dynamics estimation. *IEEE TPAMI*, 39(3):589–602, 2016.

[125] Na Wang, Haiqing Du, Yong Liu, Zheng Tang, and Jenq-Neng Hwang. Self-calibration of traffic surveillance cameras based on moving vehicle appearance and 3-d vehicle modeling. In *Proc. ICIP*, pages 3064–3068, 2018.

[126] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *Proc. CVPR*, pages 4724–4732, 2016.

[127] Philippe Weinzaepfel, Xavier Martin, and Cordelia Schmid. Human action localization with sparse spatial supervision. arXiv:1605.05197, 2016.

[128] Longyin Wen, Zhen Lei, Ming-Ching Chang, Honggang Qi, and Siwei Lyu. Multi-camera multi-target tracking with space-time-view hyper-graph. *IJCV*, 122(2):313–333, 2017.

[129] Nicolai Wojke and Dietrich Paulus. Global data association for the probability hypothesis density filter using network flows. In *Proc. ICRA*, pages 567–572, 2016.

[130] Changchang Wu. Critical configurations for radial distortion self-calibration. In *Proc. CVPR*, pages 25–32, 2014.

[131] Jiajun Wu, Tianfan Xue, Joseph J. Lim, Yuandong Tian, Joshua B. Tenenbaum, Antonio Torralba, and William T. Freeman. Single image 3D interpreter network. In *Proc. ECCV*, pages 365–382, 2016.

[132] Qi Wu, Te-Chin Shao, and Tsuhan Chen. Robust self-calibration from single image using RANSAC. In *Proc. ISVC*, pages 230–237, 2007.

[133] Yuanlu Xu, Xiaobai Liu, Yang Liu, and Song-Chun Zhu. Multi-view people tracking via hierarchical trajectory composition. In *Proc. CVPR*, pages 4256–4265, 2016.

[134] Yuanlu Xu, Xiaobai Liu, Lei Qin, and Song-Chun Zhu. Cross-view people tracking by scene-centered spatio-temporal parsing. In *Proc. AAAI*, pages 4299–4305, 2017.

[135] Bo Yang and Ram Nevatia. Multi-target tracking by online learning of non-linear motion patterns and robust appearance models. In *Proc. CVPR*, pages 1918–1925, 2012.

[136] Bo Yang and Ram Nevatia. Online learned discriminative part-based appearance models for multi-human tracking. In *Proc. ECCV*, pages 484–498, 2012.

[137] Min Yang, Yuwei Wu, and Yunde Jia. A hybrid data association framework for robust online multi-object tracking. *IEEE TIP*, 26(12):5667–5679, 2017.

[138] Wei Yang, Shuang Li, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Learning feature pyramids for human pose estimation. In *Proc. ICCV*, pages 1281–1290, 2017.

[139] Wei Yang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. In *Proc. CVPR*, pages 3073–3082, 2017.

[140] Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Proc. CVPR*, pages 1385–1392, 2011.

[141] Mingmin Zhao, Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, and Dina Katabi. Through-wall human pose estimation using radio signals. In *Proc. CVPR*, pages 7356–7365, 2018.

[142] Feng Zhou and Fernando De la Torre. Spatio-temporal matching for human pose estimation in video. *IEEE TPAMI*, 38(8):1492–1504, 2016.

[143] Hui Zhou, Wanli Ouyang, Jian Cheng, Xiaogang Wang, and Hongsheng Li. Deep continuous conditional random fields with asymmetric inter-object constraints for online multi-object tracking. *IEEE TCSVT*, 29(4):1011–1022, 2019.

[144] Xiaowei Zhou, Menglong Zhu, Spyridon Leonardos, Konstantinos G. Derpanis, and Kostas Daniilidis. Sparseness meets deepness: 3D human pose estimation from monocular video. In *Proc. CVPR*, pages 4966–4975, 2016.

[145] Xingyi Zhou, Qixing Huang, Xiao Sun, Xiangyang Xue, and Yichen Wei. Towards 3D human pose estimation in the wild: A weakly-supervised approach. In *Proc. ICCV*, pages 398–407, 2017.

[146] Yun Zhou, Jianghong Han, Xiaohui Yuan, Zhenchun Wei, and Richang Hong. Inverse sparse group lasso model for robust object tracking. *IEEE TMM*, 19(8):1798–1810, 2017.