Statistical Framework Based on the Weighted Generalized Gaussian Mixture Model : Application to Robust Point Clouds Registration and Single Target Tracking

Bingwei Ge

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Electrical and Computer Engineering) at

Concordia University

Montréal, Québec, Canada

July 2021

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Bingwei Ge**

Entitled: **Statistical Framework Based on the Weighted Generalized Gaussian Mixture Model : Application to Robust Point Clouds Registration and Single Target Tracking**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Electrical and Computer Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____ Chair
Dr. Hassan Rivaz

_____ External Examiner
Dr. Bruno Lee (BCEE)

_____ Examiner
Dr. Hassan Rivaz

_____ Supervisor
Dr. Nizar Bouguila (CIISE)

Approved by _____
Yousef R. Shayan, Chair
Department of Electrical and Computer Engineering

_____ 2021

_____
Mourad Debbabi, Dean
Gina Cody School of Engineering and Computer Science

Abstract

Statistical Framework Based on the Weighted Generalized Gaussian Mixture Model : Application to Robust Point Clouds Registration and Single Target Tracking

Bingwei Ge

Due to the introduction of the shape parameter, generalized Gaussian has better modelling capabilities than the Gaussian distribution. Therefore it is broadly used in various fields. Based on this, we established a statistical framework for the data-weighted multivariate generalized Gaussian mixture model (WMGGMM). By extending the traditional EM algorithm, we obtain the model parameters in non-closed forms and then apply an iterative method employing the fixed point equation to get the values of the components' mean and covariance. The Newton-Raphson approach is used to update the shape parameters. The complexity of the model is automatically determined by the minimum message length (MML) criterion. This thesis implements the proposed framework to two challenging tasks: point clouds registration and single-target visual tracking. The data weighting approach considers different techniques depending on the specific application's needs. In the first application, we adopt k-nearest neighbours to supply greater weight to points with high density, highlighting the main structure of the entity point cloud object while reducing noise. In the second application, the distance significance is based on the spatial kernel, which means pixels closer to the center of the target candidate ellipse have more contributions.

In the first application, WMGGMMs describe the target point cloud and the point cloud to be registered. Then the KL divergence between these two models will be used as the loss function for the stochastic optimization to obtain the best transformation model parameters, decreasing the probability of falling into a local optimum. The self-built point cloud is utilized to evaluate the performance of the algorithm on rigid registration. The results show that the algorithm significantly reduces the influence of outliers, enhances the robustness and accuracy of the algorithm, and effectively extracts the critical features of the object. In the second application, the preprocessing step

of colour compression (image segmentation) improves the adaptability of the generalized Gaussian mixture model to the data while preserving the original image information as much as possible. According to the ratio of each pixel's responsiveness to the target and background model, segmentation weights of the pixels are obtained to guide the location and size update of the target. The performance of the proposed approach is experimentally verified on a public dataset and compared with other algorithms.

Acknowledgments

Time flies. My study and research in the graduate stage are short but fruitful. At the time of parting, I am proud of studying at Concordia and graduating from the Department of Electrical and Computer Engineering. My tutors, family members, classmates, and friends have given me great support and care in this process. I extend my high respect and heartfelt thanks to everyone.

First of all, I would like to thank Professor Nizar Bouguila. When I sought a supervisor, he readily accepted me. In the past two years, he has given me careful guidance on my research work and helped me as much he could. I have benefited a lot from him, gaining professional knowledge and understanding how to do academics. His kindness and patience are a model of a teacher. Without him, I would not be able to achieve what I am today. At the same time, thanks to Dr. Fatma Najar and Professor Fan, who put forward many valuable suggestions during my thesis writing and helped me solve many difficulties. Secondly, I want to thank my parents for providing financial and spiritual support for my studies. If it were not for them, I would not have the opportunity to study at Concordia. Finally, I want to thank my classmates and friends, especially Xuanbo Su and Xiaozhou Cui, who have given me a lot of life support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Multivariate Generalized Gaussian Distribution (MGGD) has attracted widespread attention in various fields due to its outstanding feature fitting ability. Furthermore, the introduction of shape parameter makes it applicable for different situations [1]. However, the expression of its probability density function is not unique, and many references have tried to simplify it to a certain extent. For example, using a diagonal matrix to independently describe the variance of each dimension can effectively reduce the amount of calculation of matrix inversion, but this will ignore the correlation between the data dimensions [2, 3]. Intuitively, the surface of equal density will be a spherical shell in d-dimensional space rather than an ellipsoid. In the pursuit of accuracy, such simplification may lead to unacceptable results. Therefore, we will retain the complete covariance matrix and obtain distribution parameters consistent with the actual data by sacrificing some computational performance. The mixture model formed by the linear combination of a limited number of single MGGDs is an extension of GMM. It will inherit its advantages and perform effective feature extraction on global data. We assume that the data distribution is not uniform in most practical applications. Because under the premise of homogeneous distribution, the probability of the sample appearing in the feature space is almost the same, which means that it is challenging to obtain obvious statistical laws (this is also why uniform distribution is usually used as non-informative priors). The kernel method will be used for data weighting to enhance further the distinction of data distribution (making the

peak sharper) to improve the fitness of the MGGMM to the data and highlight the main features of the data. According to these requirements, the estimation equations of WMGGMM parameters need to be re-derived under the framework of the EM algorithm with different weighted methods.

Since the weighted generalized Gaussian mixture model (WMGGMM) has satisfying robustness, we will apply it to point cloud registration and single-target tracking in this thesis. In point cloud registration, the point-to-point registration method is likely to fall into local optimum, especially when there are similar structural blocks in the point cloud. The mixture model-based approach can effectively deal with the above problems through the maximization of the posterior. Nevertheless, the traditional GMM model has an unsatisfactory description of the high-density region which may be the skeleton structure of the object in the point cloud, and it is easy to be disturbed with noise [4]. WMGGMM can effectively eliminate the influence of noise and outliers by using data weighting, and it reduces the possibility of falling into local optimum through stochastic optimization based on KL divergence. In single-target tracking, WMGGMM is utilized to improve the tracking algorithm proposed in [5]. The preprocessing based on colour compression and image segmentation will modify the original colour feature distribution on the premise of preserving the image semantics to give play to the advantages of generalized Gaussian in fitting peak distribution. Still, considering the data support for mixture components, the shape parameter is used as a hyper-parameter which can be adjusted manually. The response model established by WMGGMMs takes into account the mixing coefficients of similar components in the target and background. And from this, the segmentation weights guiding the update of the target position and size are obtained.

1.2 Literature Review

1.2.1 Generalized Gaussian Mixture Model

The generalized Gaussian family originated from [6], and the related essential attributes, such as the definition of probability density, were studied in [7–9]. Analytical properties of GGD, including moments, product decompositions and characteristic functions, are given in [10]. [11, 12] indicate that GGD and spherically invariant random vectors (SIRV) have similar features under some circumstances. In image and signal processing, most statistical features of data are peak-heavier tailed. The

introduction of shape parameter makes GGD an excellent choice for such data. Therefore, GGD has aroused great interest in these applications, and parameters estimation of GGD is the primary problem. [13] compares three classical GGD parameter estimation methods for different sample sizes: momentum method (MM), maximum likelihood estimation (MLE) and momentum/Newton-step (MNS). The performance of MLE is the best when the shape parameters are small, while MNS is more suitable for the data satisfied with light trailing distribution. In [1], Pascal et al. point out that although Maronna 's conditions [14] do not apply to MGGD, its MLE exists. Based on the related work of fixed-point equations analysis in [15, 16], they obtain the MLE of the scattering matrix M when the shape parameter is between 0 and 1. [17] confirms that the fixed point iteration solution of the covariance is essentially a geodesic convexity optimization. Under this premise, the local optimal is also the global optimal. The negative log-likelihood satisfies the above conditions and requires weaker assumptions, which simplifies the calculation. Based on this, they get the structural covariance estimation of high-dimensional MGGD using a small number of samples. Besides, Farias et al. propose an explicit estimator based on the transcendental moment method for the shape parameter and construct its confidence interval [18].

The generalized Gaussian Mixture Model (GGMM) is a linear combination of some single GDDs. It has a wide range of applications, such as corpus keyword recognition [19, 20], image semantic segmentation [21–23] and visual feature clustering [24]. And its parameters estimation process is more complex than that of GDD. An intuitive and common approach approximates the distribution of the mixture model by the Monte Carlo Markov Chain (MCMC) technique. Still, this method is based on universal sampling, and the approximate process will take up a mass of computational resources [25]. Srikanth et al. proposed a variational inference framework for the generalized Gaussian mixture model [26]. In the original version, they used a one-dimensional GGD. Since the shape parameter does not have a conjugate distribution, a prior distribution is not considered. Instead, it is solved numerically using Newton-Raphson method. Their framework combines the Dirichlet process with a stick-breaking representation and feature selection model in subsequent improved versions. But when considering multivariate variables, the probability density is still a simplified function with independent dimensions [27]. However, it is challenging to apply variational inference using a complete covariance matrix because most of the parameters do not have

a conjugate prior due to introducing the shape parameter. It is feasible to apply the EM algorithm framework to GGMM. However, the introduction of shape parameter makes it impossible to obtain an analytical form by derivation of Q function. The solution of the fixed-point equation of the mixture model parameters is extended from the theory of Pascal et al.. Nevertheless, the shape parameter estimation still requires the Newton-Rapson method [28]. [29] combines the accuracy of the EM algorithm with the high efficiency of an evolutionary algorithm to create the EP2 method, i.e. the mixed parameters obtained through the random search of the particle swarm optimization (PSO) are integrated into the expectation-maximization process. This design can evaluate the shape parameters with a wide range.

1.2.2 Point Cloud Registration

The purpose of registration is to find the transformation and mapping relationship between two point clouds that make them overlap as much as possible in the same space coordinate system. Point cloud registration has wide applications in reverse engineering, robot vision SLAM positioning, medical imaging, architecture and topographic surveying [30]. In this section, we will introduce relevant works according to the fineness (coarse and fine matching) of registration.

The 4PCS (4-points congruent Sets) point cloud coarse registration method is suitable for cloud registration of scenic spots with small overlap areas or significant changes in the overlap area [31]. There is no need for pre-filtering and denoising input data, and the algorithm can complete point cloud registration quickly and accurately. It employs the RANSAC framework and reduces spatial matching operations by constructing and matching congruent four-point pairs, accelerating the registration process. Specifically, a coplanar four-point set is built in any pose's input point cloud P as well as Q. Then, the affine invariance constraint is adopted to match the corresponding point pairs that meet the conditions in the coplanar four-point set. As a result, the LCP (Largest Common Pointset) strategy will find the four-point pairs with the maximum overlap after registration, and the optimal coarse matching result is obtained. Since the 4PCS algorithm can cope with complex scenes of point cloud registration tasks, a series of improved algorithms have emerged. The Super4PCS algorithm uses an intelligent indexing strategy to reduce the computational complexity of the 4PCS algorithm from $O(n^2)$ to $O(n)$ [32]. The K-4PCS is aimed at the unreasonable

point cloud downsampling (simplification) in the original algorithm and proposes a point cloud key point detection strategy, using sparse key points instead of random sampling points [33]. Semantic-keypoint 4PCS [34] is dedicated to the registration of urban architectural scenes. The semantic key points of the buildings are extracted hierarchically, and then they are applied to replace the original random sampling points for point cloud registration. Generalized 4PCS improves the construction process for the original algorithm with four coplanar points. The construction of the coplanar four-point base is generalized, and the four points are no longer strictly restricted to co-exist on the same plane. This method dramatically improves the efficiency of point cloud registration [35].

Biber et al. proposed Normal Distributions Transform (NDT) [36] in 2003. The basic idea of the NDT algorithm is first to construct the normal distribution of multi-dimensional variables based on the reference scan data's cell division. When the transformed point has the most significant posterior probability density in this distribution, the two point clouds are the best match, and the optimal transformation parameters can be obtained. On this basis, various enhanced methods have been derived. For instance, [37] extended the above method to three-dimensional and verified it with a real data set. In [38], a fast and robust registration method combining multi-layer normal distribution transform (ML-NDT) with a feature extraction algorithm is introduced. The process reduces the influence of outliers and advances the speed of calculation. Moreover, the registration based on plane segments solves the problem of data association in 3D. Another coarse registration approach, SAMPLE Consensus Initial Alignment (SAC-IA), is offered in [39]. The algorithm extracts the normal vectors and the Fast Point Feature Histograms (FPFH) after simplifying the point cloud. The FPFH, which is more efficient, describes the local geometry around a point in the point cloud. The distance between features determines the relationship between point pairs, and then several points are randomly selected to calculate the transformation matrix.

The simple and effective Iterative Closest Point (ICP) algorithm [40] is the legend of fine registration. It is widely practiced, and many improvements have been developed. The core idea of ICP is to minimize the average Euclidean distance between the transformed points and the corresponding points in the reference data by optimizing the rigid transformation parameters. The registration result is accomplished by completing the two substeps alternately until the loss function convergence,

including finding the nearest corresponding points and solving the optimal transformation. The two-level loop is required to find the nearest corresponding point directly, so the distance threshold or KD-tree is generally used to accelerate the traversal process. For point-to-point ICP problems, SVD decomposition can be adopted to obtain the closed solution. ICP does not need segmentation and feature extraction of point clouds. When the initial value is suited, its accuracy and convergence are perfect. Still, only the distance between points is considered, and the structure information of the point cloud is not involved. The result is easily affected by the initial value and falls into the local optimum. [41] proposed a precise and fast point-to-(tangent) plane registration technique based on ICP. To find the intersection point on the target surface, Park et al. project the source control point onto the target surface and then reproject the projection point onto the normal vector of the source point. This method increases the speed and accuracy of registration in several situations. [42] comprehensively considers the point-to-point, point-to-plane and plane-to-plane strategies and forms a single probabilistic framework (Generalized-ICP), which improves the accuracy and robustness of registration. Normal Iterative Closest Point (NICP) [43] considers the normal vector and local curvature and further uses the local structure information of the point cloud. The experimental results in the paper show that the performance is better than GICP.

The authors in [44] proposed Coherent Point Drift (CPD) for rigid and non-rigid point set fine registration. The algorithm uses GMM to describe the point set and converts the registration into a probability density estimation problem. Each point in the point cloud is employed as a centroid to initialize the components of the Gaussian mixture model. The transformation parameters are fitted through maximum likelihood optimization. [45] proposed an automatic weight determination method that combines genetic algorithm and Nelder-Mead simplex method to choose noise and outlier measure weights in CPD. Experiments present that the proposed technique is more robust than the baseline. Hirose et al. used Bayesian variational inference to improve CPD [46], and the motion coherence was used as a prior to make the transformation parameters more explanatory. At the same time, the algorithm enhances the adaptability to the target rotation situation and is suitable for non-Gaussian kernels.

1.2.3 Visual Tracking

The primary purpose of visual tracking is to imitate the motion perception function of the physiological optical system. The algorithm will calculate the moving target's position in each image frame by analyzing the image sequence captured by the camera to obtain its motion parameters and adjacent Correspondence between frames [47].

We will introduce some classical methods first. Mean-shift is a method based on probability density functions, which searches for the target that always follow the direction of the rising probability gradient and iteratively converges to the local peak of the probability distribution [48]. Each pixel value represents the probability that the corresponding point in the input image belongs to the target object, which is suitable for situations where the target's colour model and background are quite different. However, it cannot solve the occlusion problem of the target and cannot adapt to changes in the shape and size of the moving target. Therefore, some upgraded mean-shift algorithms with scale adaptation have been proposed. For example, SOAMTS [49] uses the zero-order moment and Bhattacharyya coefficient between the target and candidate model for practical scale estimation and applies the estimated area and second-order central moment to predict the target width, height and direction changes adaptively. [50] proposed a mean-shift scale estimation mechanism that only relied on Hellinger distance and enhanced the discrimination of the target through the weight of the background colour histogram of the target field. At the same time, this method introduces forward-backward consistency checks and regularization to improve tracking performance. Kalman filter [51] estimates the target's position in the next frame by modelling the motion of the target. This method considers that the motion model of the object obeys the Gaussian model to predict the motion state of the target and then compares with the observation model to update the state of the motion target according to the error. Particle filtering [52] defines a similarity measure based on particle distribution statistics to determine the degree of matching between particles and targets. The search process randomly sprinkles a certain distribution of particles, calculates the similarity, and adds more particles to the possible target position in the next frame to prevent the target from being lost.

The method of correlation filtering (CF) is originated in the field of signal processing. Correlation is used to express the similarity between two signals, and convolution is usually handled to show correlation operations. Its basic idea is to find a filter template and make the following image and filter template perform a convolution operation. The area with the most prominent response is the target of prediction. The principal advantage of this approach is that it is fast, making it a mainstream method of tracking. The Minimum Output Sum of Squared Error Filter (MOSSEF) [53] is the originator of correlation filtering tracking. The convolution operation in the time domain will become a simple multiplication in the frequency domain, which significantly simplifies the calculation. [54] proposes the Kernelized Correlation Filter (KCF), which collects positive and negative samples through the circulant matrix of the area around the target. This method uses the matrix's diagonalization in Fourier space to improve tracking performance significantly and enable the algorithm to meet real-time requirements. The Discriminative Scale Space Tracker (DSST) stated in [55] gains MOSSE to cope with multi-scale changes. This work regards the tracking process as two independent sub-problems, combines grayscale and HOG features to train translational correlation filtering to detect the center of the target, and applies size filtering to extract samples at various scales. The SAMF (Scale Adaptive & Multiple Features) method [56] is based on KCF and uses multiple feature fusions, including grayscale, HOG and colour namespace (CN). A multi-scale search strategy is also employed.

In recent years, due to the innovation of hardware equipment, deep learning (DL) to automatically extract features through a non-linear multilayer network structure has made remarkable achievements in various fields. Therefore, trackers based on deep learning have also attracted widespread attention [57]. There are three main types of DL trackers: (1) Based on pre-trained deep features. Martin et al. proposed a new type of continuous equation, which combines multiple convolutional layers with different spatial resolutions through a joint learning framework named Continuous Convolution Operator (C-COT) [58]. The confidence score of the continuous domain can achieve accurate sub-pixel positioning. Later, the authors released an improved version called Efficient Convolution Operator (ECO) [59]. To deal with Discriminative Correlation Filter (DCF) 's overfitting, they proposed a factorization convolution operator that constructs the convolution filter as a linear combination of basic filters. At the same time, the sample generation model is used to

reduce the number of training samples. (2) Based on the off-line training depth feature. [60] uses a Stacked Denoise Autoencoder (SDAE) to perform unsupervised offline training on the large-scale natural image data set (Tiny Images) to obtain the characterization ability of the object. The online tracking part takes the encoding part of offline SDAE, then superimposes the sigmoid classification layer to form a classification network and uses positive and negative samples to fine-tune it. In the current frame, particle filtering is used to extract a batch of candidate patches, and the one with the highest confidence becomes the final prediction target. [61] proposed SiamFC to solve the real-time tracking of neural networks. This method applies a fully convolutional twin network for similarity learning, uses the ILSVRC dataset for target detection to train, and then extends the model from the ImageNet Video domain to other video tracking dataset domains. The online tracking process only needs to be inferred . (3) Correlation filtering integrated into a deep learning framework. For example, [62] proposed CFNet, which is similar to SiamFC. This method rewrites CF into a differentiable neural network layer and then integrates it with the feature extraction network to achieve end-to-end optimization. And the algorithm calculates a new template in each frame, combines it with the previous template using a moving average to update in real-time.

1.3 Problem Statements and Thesis Contributions

The first problem in this thesis is to derive the fixed-point equations of the weighted generalized Gaussian mixture model. The introduction of weights makes the formulas more complicated, particularly when the weight parameters are utilized as random variables. Although allowing the data to express the weighting law by itself can avoid low adaptability caused by subjective definitions, the update of weights' prior parameters directly based on the MLE will increase many calculations. Applying conjugacy only involves the current small batch of data in each round of optimization, so it is essential to find an effective conjugate prior. The second problem is how to quickly reduce the redundant components in the early stage of the automatic determination of the number of components in the mixed model. In addition, there will be a large number of multiplexed sub-blocks in the iterative process of the fixed point equation. For example, the Mahalanobis distance is calculated after the inversion of the covariance matrix. How to reasonably allocate the pre-calculation in the

actual program to accelerate the algorithm convergence is also worth considering. One of the main challenges in point cloud registration applications comes from sample noise and outliers caused by different acquisition environments, sensor noise, and imaging mechanisms. We aim to find a suitable weighting method to achieve a more robust registration by removing interference as much as possible while preserving the main characteristics of the data. [5] does not fully consider the prior mixing coefficients of the same colour components in the target and the background in the visual tracking application. At the same time, it is easy to fail to track when the target is moving quickly. We will make improvements in response to the above problems.

The main contributions of this thesis are as follows: (1) The data weights are introduced into the generalized multivariate Gaussian mixture model, and the fixed point equations under EM algorithm framework are derived. (2) The minimum message length (MML) is adopted to determine the number of components in the mixture model automatically and a negative feedback mechanism for the initial components rapid reduction is introduced.

The highlights in point cloud registration are as listed: (1) A more explanatory Bayesian process is used to update the weights, which reduces the impact of insufficient prior knowledge. (2) The KL divergence between the two mixed models is applied as a loss function to guide the stochastic optimization algorithm to update registration parameters.

The improvements and innovations in visual tracking are as follows: (1) Colour compression is introduced to improve the fitness of the WMGGMM to data, reduce noise, enhance the contrast between the target and the background, and speed up the model convergence. (2) According to the ratio of each pixel's responsiveness to the target and background model, segmentation weights of the pixels are obtained to guide the location and size update of the target. (3) The model update strategy based on the change of the ellipse aspect ratio is applied to deal with the three-dimensional rotation of the target.

1.4 Thesis Outline

In Chapter 2, the WMGGMM framework combined with stochastic optimization is proposed for point cloud registration. The KL divergence between these two mixture models is utilized as the

loss function for stochastic optimization to find the optimal parameters of the transformation model. The self-built point clouds are used to evaluate the performance of the proposed algorithm on rigid registration.

In Chapter 3, a single target tracker is proposed based on probabilistic models. Colours compression is applied as preprocessing to reduce the noise of the video frames. The spatially WMGGMM with shape hyperparameters describes the feature histograms of the aim area and context. Based on the pixel-to-model responsivity ratio to generate segmentation weights, we improved the mean-shift position and size update mechanism. And we renew the model according to the change in the aspect ratio of the ellipse. The performance of the proposed algorithm is experimentally verified on a public dataset and compared with other algorithms.

In Chapter 4, we briefly give a summary of the current work and look forward to future research directions based on the existing deficiencies.

Chapter 2

Data-Weighted Multivariate Generalized Gaussian Mixture Model: Application to Point Cloud Robust Registration

2.1 Introduction

The purpose of point cloud registration is to extract the key points or features corresponding to the target point set and the point set to be registered and find the transformation mapping relationship between two point sets [63]. This task involving image processing, data analysis and computer vision has essential applications in many practical scenarios.

For instance, point cloud registration is essential for driverless technology. Indeed, various hardware sensors, such as lidars, short-wave radars and depth-of-field cameras, could be mounted on the crew-less vehicle and point cloud registration technology can be used to fuse data collected from multiple sensors [64–66] to provide fundamental functions such as scene stitching, vehicle positioning [67], and typical scene recognition and matching for vehicle control strategies. For example, the authors in [68] proposed a framework used for unmanned vehicles based on end-to-end point cloud registration deep networks. They obtained the corresponding relationship by learned matching probabilities (LMP) among a group of candidate points related to static characteristics instead

of using existing points. Point cloud registration is also applied in medical imaging [69]. Indeed, to facilitate the diagnosis of the disease, several medical images from different instruments, such as Positron Emission Tomography (PET), Computed Tomography (CT) and Magnetic Resonance Imaging (MRI), need to be combined [70]. For example, authors in [71] improved the popular iterative Closest Point (ICP) algorithm by combining 3D scale-invariant feature transform to register 3D free-form closed surfaces (human skull models). In another work the authors in [72] used the Gaussian mixture model (GMM) with a semi-supervised EM algorithm and geometric constraints to achieve retinal image registration. Moreover, 3D reconstruction also makes extensive use of point cloud registration technology. For large buildings, for example, general scanning equipments cannot complete the whole scanning process at one time because the scanning range is limited. It requires scanning multiple parts and then splicing point clouds together [73]. In other cases, the objects to be observed may be dynamic or with complex surface characteristics. The accuracy of these features plays a crucial role in modelling and analysis; repeated scanning to build a fusion model can improve details [74].

Considering the point set's acquisition perspective, noise and outliers generated in the acquisition process, as well as the deformation and part missing of the point set caused by other factors, point cloud registration is a challenging task [63]. Various methods are proposed to enhance the robustness and accuracy of point cloud registration. In terms of pairwise registration considering only two point sets, there are three main registration categories: distance-based methods including ICP [75], Graph Matching (GM) [76], filter-based methods and probability-based methods [77].

However, most point-to-point methods are prone to fall into local optima, especially if there are some similar point structure blocks in the point set. To improve this situation, registration based on mixed models (most are GMM-based) has proven effectiveness [78–80]. The core idea is to model and describe the probability distribution of the point set using a parameterized mixed model and find the closest response of the mixture model to determine the corresponding relationship between point sets. These models perform well even if the sampling rate of two point clouds is not the same.

Nonetheless, there are two evident deficiencies when using the GMM. First, the GMM cannot describe well some non-gaussian distributions, such as the typical peak-trailing distributions in

signal processing. In the point cloud, intuitively speaking, space with dense data will carry more information. These high-density areas may represent the crucial feature structures in the point cloud, yet the GMM cannot effectively fit these high-density blocks, and its results tend to be relatively average. Secondly, GMM is easy to be disturbed by noise. Different noise levels will result in the mixture model's divergent response parameters, which could compromise the final registration accuracy [4].

The goal of this paper is to propose a point cloud registration method based on the weighted multivariate generalized Gaussian mixture model (WMGGMM) that we develop in this paper to address the difficulties above. The generalized Gaussian distribution (GGD) belongs to the family of elliptic distributions. Due to the addition of a shape parameter, it has a strong ability to describe various data distributions, particularly considering the data peak [1]. Its special cases contain Gaussian and Laplace distributions, and therefore it is widely used in feature extraction [81–83] and texture retrieval [84–86]. The mixture model of GGD also has been applied in several applications such as image processing and segmentation [2, 87, 88] and human movement recognition [3, 89].

We show that the generalized Gaussian mixture model (GGMM) is an alternative worthy choice for point cloud registration when real-time is not required too much. Although most parameters of GGMM have no closed solutions, its offers high registration accuracy and robustness. In addition, we introduce weights to reinforce the ability to pay attention to dense areas and reduce the influence of noise and outliers on the parameter estimation process. After obtaining the GGMM models for the target scene and the scene to be registered, the approximate Kullback-Leibler divergence (KLD) is computed to measure the models' difference. It will be used as a loss function to find the optimal registration parameters through the stochastic optimization algorithm.

The paper is organized as follows. Section 2.2 will tackle the WMGGMM parameters estimation. This section will also give the complete learning algorithm. Section 2.3 presents some experimental results conducted on synthetic data set to verify the algorithm's performance and robustness. It is also devoted to our approach to obtain the optimal registration parameters using stochastic optimization and the final registration effect performed on rigid transformation is presented. Conclusion and future works are finally reported in section 2.4.

2.2 Weighted Multivariate Generalized Gaussian Mixture Model

In the majority of existing works, features independency assumption is used to simplify modeling high-dimensional data which results in a distribution which is a product of one-dimensional generalized Gaussians [2, 3]. Unlike these works, we use here the PDF as defined in [1]:

$$p(x;\mu,\Sigma,\beta,m)=\frac{\Gamma\left(\frac{d}{2}\right)\beta}{\Gamma\left(\frac{d}{2\beta}\right)\pi^{\frac{d}{2}}2^{\frac{d}{2\beta}}m^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}}e^{-\frac{1}{2m^{\beta}}\left((x-\mu)^{T}\Sigma^{-1}(x-\mu)\right)^{\beta}} \tag{1}$$

where $\Gamma(\cdot)$ denotes the Gamma function, $x,\mu\in\mathbb{R}^d$ and μ is the mean vector, Σ is $d\times d$ real symmetric positive definite matrix called scatter matrix; m and $\beta>0$ are the scale and shape parameters of MGGD. The shape parameter controls the peak's sharpness and the tail's extension in the probability density function. It is worth noting that when $\beta=1$, the MGGD becomes the multivariate Gaussian distribution. If $\beta<1$, the distribution will have a sharper peak and more massive tail. However, when β tends to infinity, MGGD is close to a multivariate uniform distribution [1].

2.2.1 Data Weighting

Recent research works have shown that data weighting could improve modeling capabilities (see, for instance, [90] and references therein). Here we follow these works via an extension to GGMM-based modeling. As proposed in [90], for instance, each sample will get a corresponding weight w greater than 0. If we consider the likelihood for a specific sample x, we can write it as $p(x;\mu,\Sigma,\beta,m)^w$. This is not a probability distribution because the integral is not equal to one. But, we notice that $p(x;\mu,\Sigma,\beta,m)^w\propto p(x;\mu,\Sigma,\beta,mw^{-1/\beta})$, then we can obtain the PDF with weight as a parameter:

$$\hat{p}(x;\theta,w)=p(x;\mu,\Sigma,\beta,mw^{-1/\beta}) \tag{2}$$

where $\theta=\{\mu,\Sigma,\beta,m\}$. It can be seen that individual weight directly influences m in the PDF here. Still, in parameter estimation, the weights will affect simultaneously the scale and shape in the final mixture. Having the new distribution in equation (2) in hand we can get the K-component

mixture:

$$\widetilde{p}(x;\Theta,w)=\sum_{k=1}^{K}\pi_k p(x;\mu_k,\Sigma_k,\beta_k,m_k w^{-1/\beta_k})$$

where $\Theta=\{\pi_1,...,\pi_K,\theta_1,...,\theta_K\}$ denotes the parameter set of the model, $(\pi_1,...,\pi_K)$ are the mixing coefficients which satisfy $\pi_k>0$ and $\sum_{k=1}^{K}\pi_k=1$. $\theta_k=(\mu_k,\Sigma_k,\beta_k,m_k)$ are the parameters of the k^{th} component. Let $X=\{x_1,...,x_N\}$ represents the whole data set and $W=\{w_1,...,w_N\}$ is the weights set, then the log-likelihood function is given by:

$$L(X|\Theta,W)=\sum_{i=1}^{N}\ln\left[\sum_{k=1}^{K}\pi_k p(x_i;\mu_k,\Sigma_k,\beta_k,m_k w_i^{-1/\beta_k})\right]$$

2.2.2 MGGD with Fixed Weights

For maximum likelihood estimation, the missing variables $Z=\{z_1,...,z_N\}$ are introduced. If x_i is generated by the k^{th} component, then $z_i=k$. We assume that the weights are already known by prior knowledge, so the expected complete data log-likelihood (Q function) can be written as:

$$\begin{aligned}
Q_c\left(\Theta,\Theta^{(r)}\right)&=\mathrm{E}_{\mathrm{p}\left(Z|\mathrm{X};\mathrm{W},\mathrm{Q}^{(r)}\right)}[\ln P(X,Z;W,\Theta)]\\
&\overset{\Theta}{=}\sum_{i=1}^{N}\sum_{k=1}^{K}\eta_{ik}\Bigg[\ln\pi_k+\ln\beta_k+\frac{d}{2\beta_k}\ln w_i\\
&-\ln\Gamma\left(\frac{d}{2\beta_k}\right)-\frac{d}{2\beta_k}\ln 2-\frac{d}{2}\ln m_k-\frac{1}{2}\ln|\Sigma_k|\\
&-\frac{w_i\left[(x_i-\mu_k)^T\Sigma_k^{-1}(x_i-\mu_k)\right]^{\beta_k}}{2m_k^{\beta_k}}\Bigg]
\end{aligned}\tag{3}$$

where $E_P[\cdot]$ is the expectation with respect to the distribution P and $\overset{\Theta}{=}$ represents items related only to Θ. Subsequently, the optimal parameters Θ^* can be obtained by EM algorithm. In the expectation step, the posteriors are updated with:

$$\eta_{ik}=p\left(z_i=k\mid x_i;\Theta,w_i\right)=\frac{\hat{p}\left(x_i;\theta_k,w_i\right)}{\sum_{k=1}^{K}\pi_k\hat{p}\left(x_i;\theta_k,w_i\right)}\tag{4}$$

By taking the derivatives of complete data log-likelihood with respect to parameters and making the results equal to zero, we can obtain the parameters update formulas:

$$\pi_k = \frac{1}{N}\sum_{i=1}^{N}\eta_{ik} \tag{5}$$

$$m_k = \left[\frac{\beta_k}{d\sum_{i=1}^{N}\eta_{ik}}\sum_{i=1}^{N}\eta_{ik}w_i\left[(x_i-\mu_k)^T\Sigma_k^{-1}(x_i-\mu_k)\right]^{\beta_k}\right]^{\frac{1}{\beta_k}} \tag{6}$$

$$\mu_k = \frac{\sum_{i=1}^{N}\eta_{ik}w_i\left[(x_i-\mu_k)^T\Sigma_k^{-1}(x_i-\mu_k)\right]^{\beta_k-1}x_i}{\sum_{i=1}^{N}\eta_{ik}w_i\left[(x_i-\mu_k)^T\Sigma_k^{-1}(x_i-\mu_k)\right]^{\beta_k-1}} \tag{7}$$

$$\Sigma_k = \frac{\beta_k}{m_k^{\beta_k}\sum_{i=1}^{N}\eta_{ik}}\sum_{i=1}^{N}\left[\frac{\eta_{ik}w_i(x_i-\mu_k)(x_i-\mu_k)^T}{\left[(x_i-\mu_k)^T\Sigma_k^{-1}(x_i-\mu_k)\right]^{1-\beta_k}}\right] \tag{8}$$

After replacing m_k in equation (8) with the previous result in equation (6), we can find the Σ_k is independent form m_k. If we let $y_i = (x_i-\mu_k)^T\Sigma_k^{-1}(x_i-\mu_k)$, we have:

$$\Sigma_k = \frac{d}{\sum_{i=1}^{N}\left(\eta_{ik}w_i y_i^{\beta_k}\right)}\sum_{i=1}^{N}\left[\frac{\eta_{ik}w_i(x_i-\mu_k)(x_i-\mu_k)^T}{y_i^{1-\beta_k}}\right] \tag{9}$$

And it is worth noting that μ_k and Σ_k do not have closed solutions; they are both solved through the fixed point (FP) method. According to Banach fixed point theorem [91], if (S,d) is a non-empty complete metric space with a contraction mapping $T: S \to S$, then there exists a unique fixed point S^* in S. The authors in [1] proved the convergence of Σ_k and explained the existence and uniqueness of the fixed point, based on the fact that $\beta \in (0,1]$ and Σ is a positive definite real symmetric matrix. This is also consistent with our assumptions about β. We introduce the Frobenius norm defined in equation (10) to measure the difference between S_n and S_{n-1} in the fixed point iteration. The process will stop when the approximate solution satisfies preset precision. Besides, to ensure the convergence of the FP equation, the value range of weights should also be between zero and one.

$$\|S_n - S_{n-1}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}\left|s_n^{ij} - s_{n-1}^{ij}\right|^2} \tag{10}$$

Then parameter β can be estimated using Newton-Raphson iterations [3, 89, 90]:

$$\beta_k^{(\mathrm{t}+1)} = \beta_k^{(\mathrm{t})} - \xi \frac{f\left(\beta_k^{(\mathrm{t})}\right)}{f'\left(\beta_k^{(\mathrm{t})}\right)} \tag{11}$$

where ξ is the learning rate used to prevent the oscillation and overflow in the iterative process, and it is usually around 0.1. If necessary, the method of exponential decay can be applied to make the convergence more stable. $f(\beta_k^{(\mathrm{t})})$ and $f'(\beta_k^{(\mathrm{t})})$ are given as follows:

$$\begin{aligned} f\left(\beta_k\right) = & \frac{d\sum_{i=1}^{N}\eta_{ik}}{2\sum_{i=1}^{N}\left(\eta_{ik}w_i y_i^{\beta_k}\right)}\sum_{i=1}^{N}\left[\eta_{ik}w_i y_i^{\beta_k}\ln\left(y_i\right)\right] \\ & + \frac{d\sum_{i=1}^{N}\left(\eta_{ik}\ln w_i\right)}{2\beta_k} - \frac{d\sum_{i=1}^{N}\eta_{ik}[}{2\beta_k}\left[\Psi\left(\frac{d}{2\beta_k}\right)+\ln 2\right] \\ & - \sum_{i=1}^{N}\eta_{ik} - \frac{d\sum_{i=1}^{N}\eta_{ik}}{2\beta_k}\ln\left[\frac{\beta_k}{d\sum_{i=1}^{N}\eta_{ik}}\sum_{i=1}^{N}\left(\eta_{ik}w_i y_i^{\beta_k}\right)\right] \end{aligned} \tag{12}$$

$$\begin{aligned} f'\left(\beta_k\right) = & \frac{d\sum_{i=1}^{N}\eta_{ik}}{2\left(\sum_{i=1}^{N}\eta_{ik}w_i y_i^{\beta_k}\right)^2}\left[\left(\sum_{i=1}^{N}\eta_{ik}w_i y_i^{\beta_k}\right)\right. \\ & \left.\left(\sum_{i=1}^{N}\eta_{ik}w_i y_i^{\beta_k}\left(\ln y_i\right)^2\right) - \left(\sum_{i=1}^{N}\eta_{ik}w_i y_i^{\beta_k}\ln y_i\right)^2\right] \\ & - \frac{d\sum_{i=1}^{N}\left(\eta_{ik}\ln w_i\right)}{2\beta_k^2} + \frac{d\sum_{i=1}^{N}\eta_{ik}}{2\beta_k^2}\left[\Psi\left(\frac{d}{2\beta_k}\right)+\ln 2\right] \\ & + \frac{\mathrm{d}^2\sum_{i=1}^{N}\eta_{ik}}{4\beta_k^3}\Psi'\left(\frac{d}{2\beta_k}\right) + \frac{d\sum_{i=1}^{N}\eta_{ik}}{2\beta_k^2}\left[\ln\left(\frac{\beta_k}{dN}\right)\right. \\ & \left. + \ln\left(\sum_{i=1}^{N}\eta_{ik}w_i y_i^{\beta_k}\right) - 1 - \beta_k\frac{\sum_{i=1}^{N}\eta_{ik}w_i y_i^{\beta_k}\ln y_i}{\sum_{i=1}^{N}\eta_{ik}w_i y_i^{\beta_k}}\right] \end{aligned} \tag{13}$$

where $\Psi(\cdot)$ is the digamma function.

2.2.3 Weights Considered as Random Variables

Above, we have derived the WMGGMM parameter updating formulas with fixed weights. However, it is pointed out in [90] that the Bayesian formalism is more inclined to treat parameters as

random variables and update the posterior of parameters by combining the parameters prior with the observed sample. Under this framework, the limitation of insufficient prior knowledge on the accuracy of weights will be reduced, and the inference process of weights will also be more explanatory. As mentioned before, the generalized Gaussian distribution belongs to the family of elliptic distributions so we select the same prior distribution, i.e. Gamma distribution, as in [90]. At this point, the prior and the posterior of parameters have the same form. The advantage is when we make a new observation, we do not have to re-calculate the whole process but only directly obtain the posterior distribution through the parameters, which undoubtedly simplifies the updating process of weight parameters. Then, the posterior will become the prior in the next calculation. Therefore, we can make:

$$p(w;\phi) = G(w;a,b) = \Gamma(a)^{-1} b^a w^{a-1} e^{-bw} \tag{14}$$

where $G(w;a,b)$ is the Gamma distribution, and $\phi = \{a,b\}$ are the parameters of the prior distribution of w. The mean and variance of random variable w are given by:

$$E[w] = a/b \tag{15}$$

$$Var[w] = a/b^2 \tag{16}$$

Due to the addition of prior parameters, the log-likelihood of complete data becomes the following form:

$$Q_c\left(\Theta, \Theta^{(r)}\right) = \mathrm{E}_{p\left(Z,\mathrm{W}|\mathrm{X};\Theta^{(r)},\Phi\right)}[\ln P(X,Z,W;\Theta,\Phi)] \tag{17}$$

where $\Phi = \{\phi_1, ..., \phi_N\}$ and $\phi_i = \{a_i, b_i\}$. The posterior distribution factorizes on i is shown as follows:

$$P\left(Z,W \mid X;\Theta^{(r)},\Phi\right) = \prod_{i=1}^{N} p\left(z_i, w_i \mid x_i;\Theta^{(r)},\phi_i\right) \tag{18}$$

Each of these product terms can be expressed as two-factor expressions:

$$p\left(z_i, w_i \mid x_i;\Theta^{(r)},\phi_i\right) = p\left(w_i \mid z_i, x_i;\Theta^{(r)},\phi_i\right) p\left(z_i \mid x_i;\Theta^{(r)},\phi_i\right) \tag{19}$$

According to the above formula, the expectation step in the EM algorithm is divided into two parts (E-Z step and E-W step). In the E-Z step, the marginal posterior distribution of z_i is obtained by integrating over w_i:

$$\begin{aligned}
\eta_{ik} &= \int p\left(z_i = k, w_i \mid x_i; \Theta^{(r)}, \phi_i\right) dw_i \\
&\overset{k}{\propto} \int \pi_k p\left(x_i \mid z_i = k, w_i; \Theta^{(r)}\right) p\left(w_i; \phi_i\right) dw_i \\
&= \int \pi_k \hat{p}\left(x_i; \theta_k, w_i\right) G\left(w_i; a_i, b_i\right) dw_i \\
&\propto \pi_k \bar{p}\left(x_i; \mu_k, \Sigma_k, \beta_k, m_k, a_i, b_i\right)
\end{aligned} \tag{20}$$

where $\bar{p}\left(x; \mu, \Sigma, \beta, m, a, b\right)$ is given as:

$$\bar{p}(x; \mu, \Sigma, \beta, m, a, b) = \frac{\beta \Gamma\left(\frac{d}{2}\right) \Gamma\left(a + \frac{d}{2\beta}\right)}{(m\pi)^{\frac{d}{2}} (2b)^{\frac{d}{2\beta}} |\Sigma|^{\frac{1}{2}} \Gamma(a) \Gamma\left(\frac{d}{2\beta}\right)} \left[\frac{\left((x-\mu)^T \Sigma^{-1} (x-\mu)\right)^{\beta}}{2bm^{\beta}} + 1\right]^{-\left(a + \frac{d}{2\beta}\right)} \tag{21}$$

In step E-W, according to the property of conjugate distribution, we can get:

$$\begin{aligned}
&p\left(w_i \mid z_i = k, x_i; \Theta, \phi_i\right) \overset{w_i}{\propto} p\left(x_i \mid z_i = k, w_i; \Theta\right) p\left(w_i; \phi_i\right) \\
&= \hat{p}\left(x_i; \theta_k, w_i\right) G\left(w_i; a_i, b_i\right) = G\left(w_i; a_i^{(r+1)}, b_i^{(r+1)}\right)
\end{aligned} \tag{22}$$

Thus, the updating formulas of prior parameters can be obtained:

$$a_{ik}^{(r+1)} = a_i^{(0)} + \frac{d}{2\beta_k} \tag{23}$$

$$b_{ik}^{(r+1)} = b_i^{(0)} + \frac{\left[(x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k)\right]^{\beta_k}}{2m_k^{\beta_k}} \tag{24}$$

$$\bar{w}_{ik} = \mathrm{E}_{P(w_i | z_i = k, x_i, \Theta^{(r)}, \phi_i)}[w_i] = \frac{a_{ik}^{(r+1)}}{b_{ik}^{(r+1)}} \tag{25}$$

The posterior weight mean is given as [90]:

$$\bar{w}_i^{(r+1)} = \mathrm{E}[w_i | x_i; \Theta^{(r)}, \phi_i] = \sum_{k=1}^{K} \eta_{ik}^{(r+1)} \bar{w}_{il}^{(r+1)} \tag{26}$$

This explains the outliers shielding feature of the weighted algorithm. As an outlier is by definition far from the centers of all components, it has a low posterior weight $\bar{w}_{ik}$ for each component and then a low mean posterior probability $\bar{w}_i$. By expanding equation (17), we can get a result similar to the Q function with fixed weights:

$$\begin{aligned} Q_c\left(\Theta, \Theta^{(r)}\right) &= \sum_{i=1}^{N}\sum_{k=1}^{K}\int_{w_i} \eta_{ik} \ln \pi_k \hat{p}\left(x_i; \theta_k, w_i\right) \\ &\times p\left(w_i \mid x_i, z_i = k, \Theta^{(r)}, \phi_i\right) dw_i \\ &\overset{\Theta}{=} \sum_{i=1}^{N}\sum_{k=1}^{K}\eta_{ik}\Bigg[\ln \pi_k + \ln \beta_k + \frac{d}{2\beta_k}\ln \bar{w}_{ik}^{(r+1)} \\ &- \ln \Gamma\left(\frac{d}{2\beta_k}\right) - \frac{d}{2\beta_k}\ln 2 - \frac{d}{2}\ln m_k - \frac{1}{2}\ln|\Sigma_k| \\ &- \frac{\bar{w}_{ik}^{(r+1)}\left[\left(x_i-\mu_k\right)^T \Sigma_k^{-1}\left(x_i-\mu_k\right)\right]^{\beta_k}}{2m_k^{\beta_k}}\Bigg] \end{aligned} \tag{27}$$

Therefore, w_i will be replaced by $\bar{w}_{ik}$ in all the parameters updates formulas of the mixture model. Since the equations are very similar, they are not repeated here.

2.2.4 Automatic Determination of the Number of Components

The model selection problem is tackled using the minimum message length (MML) criterion as proposed in [92]:

$$\Theta_{MML} = \arg\min_{\Theta}\left\{-\log P(\Theta) - Q_R\left(\Theta, \Theta^{(r+1)}\right) + \frac{1}{2}\log|I_c(\Theta)| + \frac{D(\Theta)}{2}\left(1+\log\frac{1}{12}\right)\right\} \tag{28}$$

where $I_C(\Theta)$ denotes the expected complete Fisher information matrix (FIM) and $D(\Theta)$ is the dimensionality of the model. Using a similar development as in [92], we can show that

$$\Theta_{MML} = \arg\min_{\Theta}\left\{\frac{M}{2}\sum_{k\in\mathbf{K}^+}\log \pi_k - Q_R\left(\Theta, \Theta^{(r+1)}\right) + \frac{K^+(M+1)}{2}\left(1+\log\frac{n}{12}\right)\right\} \tag{29}$$

where $\mathbf{K}^+$ is the set of non-empty components and K^+ is the number of elements in it. Moreover, we can rewrite the formula for calculating π_k in maximization step of EM algorithm:

$$\pi_k = \frac{\max\left(0, \sum_{i=1}^{N} \eta_{ik} - MK^+/2\right)}{\sum_{l=1}^{K} \max\left(0, \sum_{i=1}^{N} \eta_{il} - MK^+/2\right)} \tag{30}$$

The threshold for minimum support is high when the number of non-empty components is large at the beginning. Some components can be removed quickly. In the process of component updating one by one, the threshold gradually approaches the situation in formula (30) as the number of non-empty components decreases.

2.2.5 Complete Algorithm

The complete steps are outlined in Algorithm 1. We use k-means to initialize π_k, μ_k, Σ_k. The initial parameter β_k is specified as 0.5, and the parameter m_k is calculated according to the pre-clustering results and the formula in [1]:

$$m_k = \left\{ \frac{\beta_k}{dN_k} \sum_{i=1}^{N_k} \left[(x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k) \right]^{\beta_k} \right\}^{\frac{1}{\beta_k}} \tag{31}$$

where $x_i \in X_k$ is the i_{th} sample for the k_{th} cluster and N_k is the number of samples. We adopt the same data similarity measurement method based on the Gaussian kernel as in [90] for the initialization of weights. However, due to the constraint of the weight range, we modify it as follows:

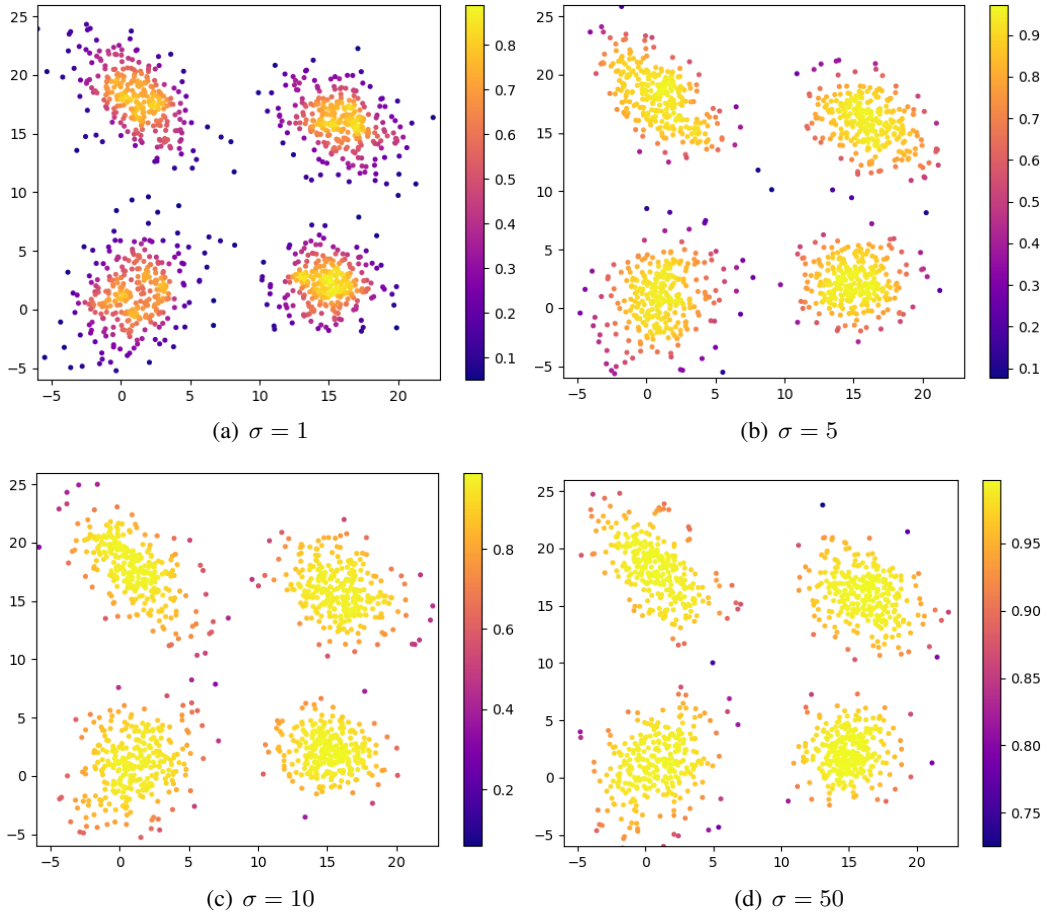
$$w_i = \frac{1}{q} \sum_{j \in S_i^q} exp\left[-\frac{d^2(x_i, x_j)}{\sigma} \right] \tag{32}$$

where $d^2(x_i, x_j)$ denotes the euclidean distance and S_i^q is the set containing q nearest neighbors of x_i, σ is a positive scale. The default setting for q is 20. Then we can calculate the initialization of the prior parameters of the weight through equations (15) and (16), i.e. $a_i = w_i^2$ and $b_i = w_i$. Fig.2.1 shows the impact of different σ on weights; its significance is to change the degree of differentiation of weights. When there is a small σ, the difference in weights between dense and sparse areas becomes more pronounced. Conversely, the distribution of weights will be relatively close. In other

Algorithm 1 Proposed WMGGMM Algorithm with Component-Wise EM Procedure

Require: $X = \{x_i\}_{i=1}^{N}$; $\Phi^{(0)} = \{a_i^{(0)}, b_i^{(0)}\}_{i=1}^{N}$; K_{max}
$\Theta^{(0)} = \{\pi_k^{(0)}, \mu_k^{(0)}, \Sigma_k^{(0)}, \beta_k^{(0)}, m_k^{(0)}\}_{k=1}^{K_{max}}$;
Ensure: Optimal mixture model parameters: Θ^*
Set: $r = 0$, $MML = +\infty$
repeat
for $k = 1$ To K_{max} **do**
E-Z step using (20):
$\eta_{ik}^{(r+1)} = \frac{\pi_k^{(r)} \overline{p}(x_i;\mu_k^{(r)},\Sigma_k^{(r)},\beta_k^{(r)},m_k^{(r)},a_{ik}^{(r)},b_{ik}^{(r)})}{\sum_{l=1}^{k_{max}} \pi_l^{(r)} \overline{p}(x_i;\mu_l^{(r)},\Sigma_l^{(r)},\beta_l^{(r)},m_l^{(r)},a_{il}^{(r)},b_{il}^{(r)})}$
Compute the # of non-empty components: K^+
E-W step using (23) - (25):
$a_{ik}^{(r+1)} = a_i^{(0)} + \frac{d}{2\beta_k}$
$b_{ik}^{(r+1)} = b_i^{(0)} + \frac{[(x_i-\mu_k)^T \Sigma_k^{-1}(x_i-\mu_k)]^{\beta_k}}{2m_k^{\beta_k}}$
$\overline{w}_{ik} = a_{ik}^{(r+1)} / b_{ik}^{(r+1)}$
M step:
$\pi_k^{(r+1} = \frac{MAX(0,\eta_{ik}^{(r+1)} - K^+ M/2)}{\sum_{l=1}^{K_{max}} MAX(0,\eta_{il}^{(r+1)} - K^+ M/2)}$
if $\pi_k^{(r+1)} > 0$ **then**
Update μ_k using (7):
repeat
$\mu_k^{new} = T(\mu_k^{old})$
until $\|\mu_k^{new} - \mu_k^{old}\|_F < \epsilon$
$\mu_k^{(r+1)} = \mu_k^{new}$
Update Σ_k using (8):
repeat
$\Sigma_k^{new} = T(\Sigma_k^{old})$
until $\|\Sigma_k^{new} - \Sigma_k^{old}\|_F < \epsilon$
$\Sigma_k^{(r+1)} = \Sigma_k^{new}$
Update β_k using (11)-(13):
repeat
$\beta_k^{new} = \beta_k^{old} - \zeta \frac{f(\beta_k^{old})}{f'(\beta_k^{old})}$
until $|\mu_k^{new} - \mu_k^{old}| < \epsilon$
$\beta_k^{(r+1)} = \beta_k^{new}$
Update $m_k^{(r+1)}$ using (6)
end if
end for
Compute $MML^{(r+1)}$ using (29)
$r = r + 1$
until $|\Delta MML^{(r)}| < \epsilon$
Return the parameters Θ of non-empty components as optimal mixture model parameters Θ^*

words, a smaller σ is more beneficial at removing outliers and noise. But it is important to note that when the σ is too small, the weight operation is equivalent to remove most of the points that away from the clustering center in the data distribution. The actual points involved in parameter identification are reduced, leading to inadequate support of components. Also, the resulting mixed model parameters will have a large deviation from the original data distribution. Therefore, the selection of σ is a balance between model robustness and model accuracy. We set it to 25 in our cases.



(a) $\sigma = 1$ (b) $\sigma = 5$

(c) $\sigma = 10$ (d) $\sigma = 50$

Figure 2.1: The influence of different Gaussian kernel parameter σ on the weights.

2.3 Experimental Results

2.3.1 Synthetic Data

First, we demonstrate the performance of the WMGGMM model through synthetic data. The method for generating data points that follow a MGGD distribution comes from [1]:

$$\mathbf{x} \in \mathbb{R}^d \widetilde{=} \tau C^{1/2} \mathbf{u} \tag{33}$$

where $\mathbf{x}$ is a random vector that follows a MGGD with scatter matrix $C = m\Sigma$ and shape parameter β, and $\widetilde{=}$ denotes equivalent on distribution; $\mathbf{x}$ is a random vector uniformly distributed on a unit sphere and τ is positive scalar random variable that satisfies $\tau^{2\beta} \sim \Gamma\left(d/(2\beta), 2\right)$. Table 2.1 and Table 2.2 show the parameter estimation results for two generated two-dimensional data sets from 4- and 3-component mixture models, respectively, where $\rho_k = cov(X,Y)/\sqrt{var(X)var(Y)}$ represents the correlation coefficient used to measure the slope of the scatter matrix. We can see that the estimated parameters are close to the real ones. According to the correlation coefficient comparisons, the slopes of the real and estimated scatter matrices are also close. Due to the role of weight, the points essentially involved in parameter estimation are concentrated in data-intensive areas, leading to reduced scale parameter m. However, the change in parameter β is not necessarily a decrease. Suppose that β is small in the default setting. In the big trailing part, the data points are distributed sparsely. The weight operation will remove most of these points, so the shape of data is changed, and the final estimated β will become larger instead.

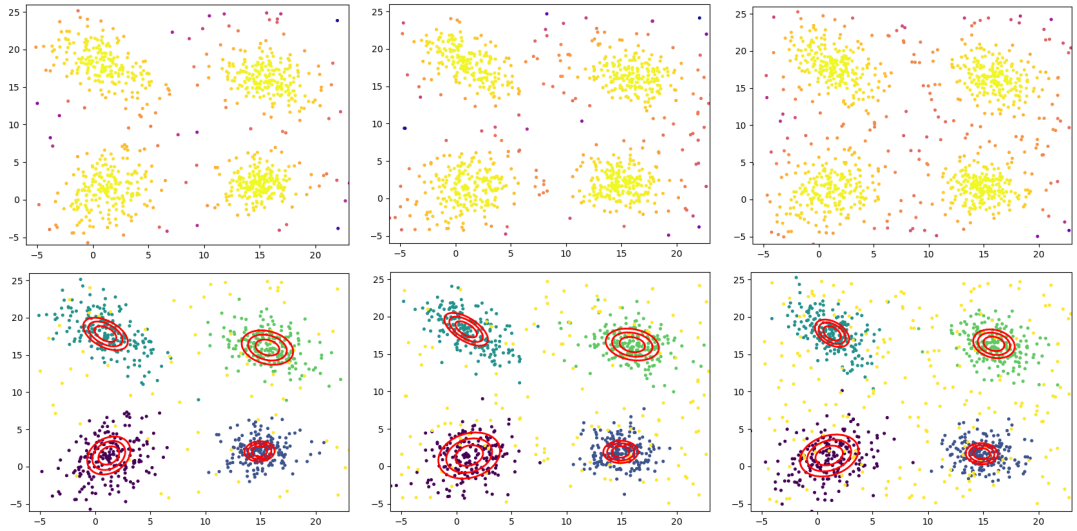
Fig.2.2 presents the parameter estimation results at different noise levels. Parameter estimation is carried out after the proportional addition of random uniform noise in the primary distribution area ([-5,25;-5,25]) of the original data. When σ is appropriately selected, the weights remove most of the noise and outliers, and the actual points involved in parameter estimation are mostly in the desired original data distribution. The final results show that the mixture models of the three different cases are very similar, which effectively proves the WMGGMM algorithm's robustness. The results of the mixture model for overlapping data are shown in Fig.2.3. The introduction of shape parameter β enables MGGD to improve the identification ability of data distribution peaks.

Table 2.1: Estimation of the mixture model parameters for the first synthetic data set.

Default mixture model parameters				
π_k	μ_k	β_k	C_k	ρ_k
0.25 (N=300)	$\begin{bmatrix}1 & 1\end{bmatrix}$	0.85	$\begin{bmatrix}3 & 1\\ 1 & 5\end{bmatrix}$	0.35
0.25 (N=300)	$\begin{bmatrix}15 & 2\end{bmatrix}$	0.85	$\begin{bmatrix}2 & 0\\ 0 & 2\end{bmatrix}$	0.00
0.25 (N=300)	$\begin{bmatrix}1 & 18\end{bmatrix}$	0.85	$\begin{bmatrix}3 & -2\\ -2 & 4\end{bmatrix}$	-0.58
0.25 (N=300)	$\begin{bmatrix}16 & 16\end{bmatrix}$	0.85	$\begin{bmatrix}3 & -1\\ -1 & 3\end{bmatrix}$	-0.33
Estimated mixture model parameters				
π_k	μ_k	β_k	C_k	ρ_k
0.2444	$\begin{bmatrix}0.98 & 0.90\end{bmatrix}$	0.72	$\begin{bmatrix}1.43 & 0.57\\ 0.57 & 2.94\end{bmatrix}$	0.28
0.2586	$\begin{bmatrix}15.11 & 2.05\end{bmatrix}$	0.70	$\begin{bmatrix}0.60 & -0.04\\ -0.04 & 0.67\end{bmatrix}$	-0.06
0.2482	$\begin{bmatrix}1.15 & 17.99\end{bmatrix}$	0.75	$\begin{bmatrix}1.39 & -1.00\\ -1.00 & 2.15\end{bmatrix}$	-0.58
0.2486	$\begin{bmatrix}15.98 & 15.99\end{bmatrix}$	0.77	$\begin{bmatrix}1.12 & -0.3\\ -0.3 & 1.05\end{bmatrix}$	-0.27

Table 2.2: Estimation of the mixture model parameters for the second synthetic data set.

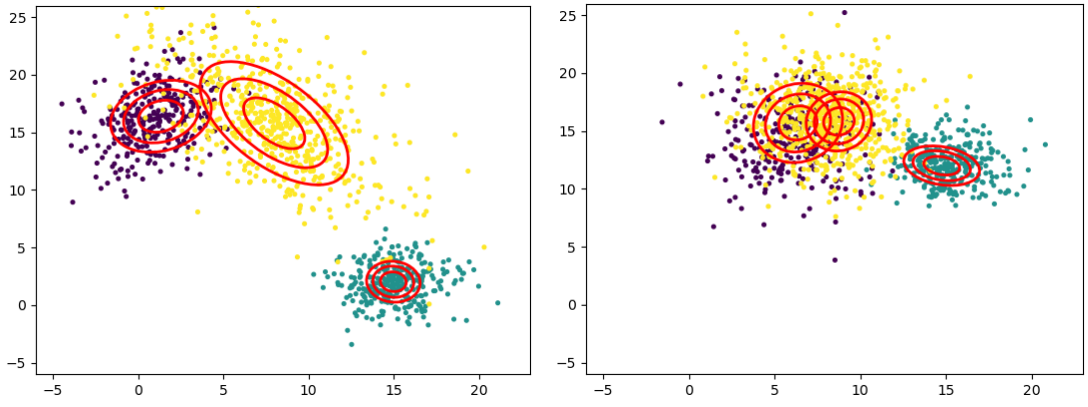
Default mixture model parameters				
π_k	μ_k	β_k	C_k	ρ_k
0.25 (N=300)	$\begin{bmatrix}8 & 16\end{bmatrix}$	0.60	$\begin{bmatrix}3 & -2\\ -2 & 4\end{bmatrix}$	-0.58
0.25 (N=300)	$\begin{bmatrix}15 & 2\end{bmatrix}$	0.85	$\begin{bmatrix}2 & 0\\ 0 & 2\end{bmatrix}$	0.00
0.50 (N=600)	$\begin{bmatrix}1 & 3\end{bmatrix}$	0.85	$\begin{bmatrix}3 & 1\\ 1 & 5\end{bmatrix}$	0.35
Estimated mixture model parameters				
π_k	μ_k	β_k	C_k	ρ_k
0.2596	$\begin{bmatrix}7.89 & 16.05\end{bmatrix}$	0.76	$\begin{bmatrix}2.27 & -1.34\\ -1.34 & 3.39\end{bmatrix}$	-0.48
0.2817	$\begin{bmatrix}14.90 & 2.12\end{bmatrix}$	0.67	$\begin{bmatrix}0.62 & -0.17\\ -0.17 & 0.78\end{bmatrix}$	-0.24
0.4587	$\begin{bmatrix}0.98 & 3.11\end{bmatrix}$	0.70	$\begin{bmatrix}0.91 & 0.61\\ 0.61 & 2.12\end{bmatrix}$	0.44



(a) 10% noise (b) 20% noise (c) 30% noise

Figure 2.2: Estimation results of mixture model parameters with different noise levels (10%, 20%, 30%).

Although the overall distribution of two components overlaps, the WMGGMM algorithm can still accurately identify the parameters of each component when their centers do not coincide. Fig.2.4



(a) (b)

Figure 2.3: The result of parameter estimation in the case of components overlap.

displays the execution time of the WMGGMM algorithm under different conditions (the default number of components is 4). The average value of 5 independent running times is taken as the final result, and the unit is seconds. Because most of the algorithm's parameters need to be estimated iteratively (fixed-point equations and Newton-Raphson method), the algorithm's running time is

much higher than that of the standard EM algorithm. The anti-interference performance of the model is improved by sacrificing some efficiency. The time here can only be used as a relative reference since algorithm run time is influenced by computer performance, hardware and software acceleration, and randomness of parameter initialization.

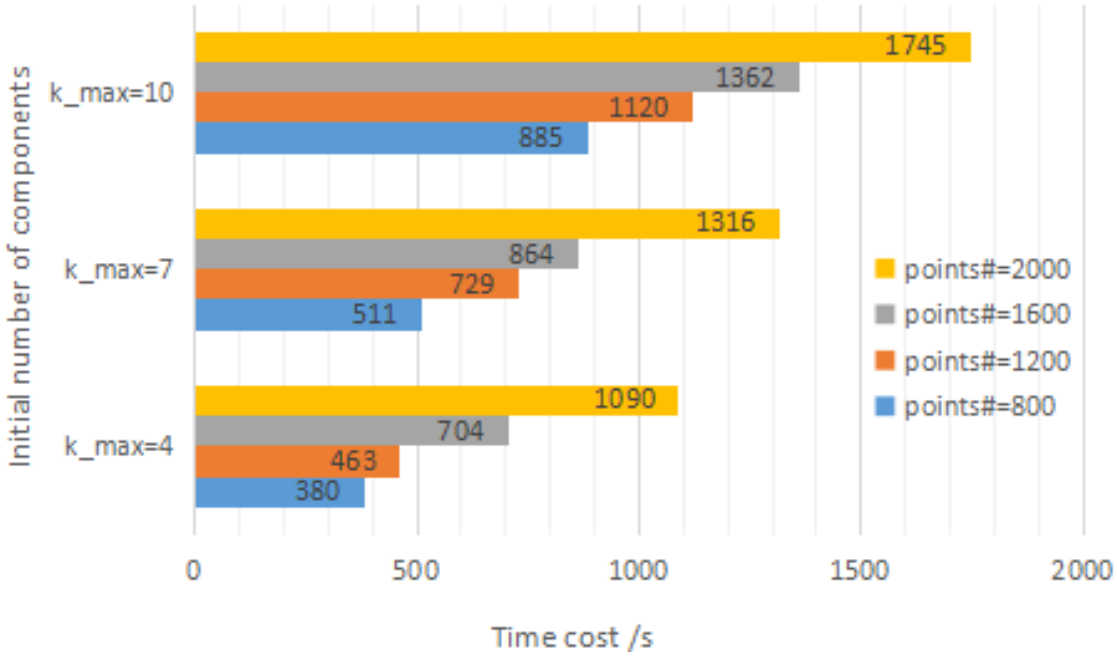


Figure 2.4: Performance analysis of WMGGMM for 2-D data

2.3.2 Point Cloud Registration Using WMGGMM

We consider the approach proposed in [78], in the case of GMM, to perform WMGGMM-based registration. Two mixture models are adopted to describe the target scene and the scene to be registered. Then, the difference between these two models is applied to update the registration parameters. Nevertheless, the algorithm proposed in [78] is still a method based on point to point in essence. The data pre-clustering is not considered, but each point in the data set is used as a mean to initialize a mixture component. This approach is equivalent to converting the entire data set into a mixture model with multiple simple components. However, this method of initializing the mixture model is not suitable in the point cloud with large data volume. The subsequent L2 divergence provides convenience for the derivation based on the transformation model. But, the derivative

optimization process is closely related to the expression of the specific model. If the transformation model is changed, all optimization processes need to be rederived. Therefore, we propose a method based on WMGGMM used to pre-cluster the target scene and the scene to be registered and extract the key features to form the mixture models. Then, the optimal registration parameters are found by using stochastic optimization through the KLD between the mixture models.The KLD of the two statistical models is defined as follows:

$$KLD(f||g) = \int_x f(x) \log \frac{f(x)}{g(x)} dx \tag{34}$$

Unfortunately, the KLD between mixture models has no analytical expression, so several approximations have been proposed [93–95]. Compared with GMM, the KLD of generalized Gaussian mixture model using the inner product of components is too complicated due to the introduction of shape parameters. Even though a method for calculating the KLD of two MGGDs is proposed in [86], the matched bound and variational approximations are also not feasible because the premise of this approach is that all MGGDs have zero mean. However, the mean value of components is significant in the point cloud registration scenario. Therefore, we finally used Monte Carlo sampling to calculate the KLD between two GGMMs:

$$KLD_{MC}(P_S||P_M) = \frac{1}{n} \sum_{i=1}^{n} [\log P_S(x_i) - \log P_M(x_i)] \tag{35}$$

where $P_S(x)$ and $P_M(x)$ are the probability density functions of GGMM obtained from the target scene and the scene to be registered, respectively, and $\{x_i\}_{i=1}^n$ denotes n samples taken from $P_S(x)$. In the above formula, the sum of the probability of samples replaces the integral, and the Monte Carlo method is the only method that can approximate the actual value of KLD when there are enough sample points. The Gibbs sampling, one of the Markov Chain Monte Carlo (MCMC) sampling techniques, is applied in this process using 1000. We assume that the point cloud transformation model is:

$$X = f_T(X_0, \Omega) \tag{36}$$

where X_0 and X are the point sets before and after the transformation, $\Omega = \{\omega_1, ..., \omega_m\}$ is the parameter set of the transformation model. Then, we can express the point cloud registration problem in the following form:

$$\Omega^* = \arg\min_{\Omega} \{KLD_{MC}\left[\text{ggmm}(\mathbf{S}) \| \text{ggmm}\left(f_T(\mathbf{M}, \Omega)\right)\right]\} \tag{37}$$

where $\mathbf{S}$ is the target scene point set and $\mathbf{M}$ is the point set to be registered, $ggmm(\cdot)$ denotes the obtained PDF of GGMM from a data set using the WMGGMM algorithm. The complete point cloud registration algorithm using WMGGMM is exhibited in Algorithm 2. However, it is only a general framework and does not specify a concrete random optimization method. The stochastic optimization technique can be selected depending on needs, as long as the definition of the loss function in (37) is satisfied, and the optimization domain is Ω. The simulated annealing (SA) algorithm is used in this paper. When the transformation model is rigid, i.e. $X = R_\alpha X_0 + t$, the

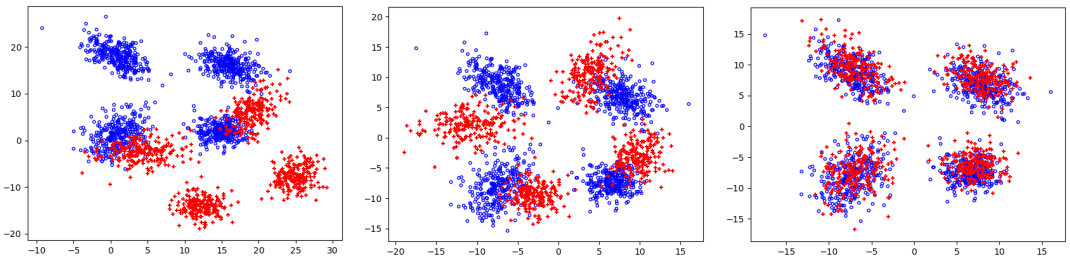
Algorithm 2 Point Cloud Registration Algorithm Based on WMGGMM and Stochastic Optimization

Require: Scene set: S, Model set: M, Initial transformation parameter set: $\Omega^{(0)}$
Ensure: Optimal transformation parameter set: Ω^*
  Set: $KLD^{old} = 0$, $KLD^{new} = 0$, $\Omega^{old} = \Omega^{(0)}$
  $ggmm_s = GGMM(S)$
  Gibbs sampling on $ggmm_s$ to get: $X_{sample} = \{x_i\}_{i=1}^{1000}$
  repeat
    $ggmm_m^{old} = ggmm(f_T(M, \Omega^{old}))$
    $KLD^{old} = KLD_{MC}(ggmm_s || ggmm_m^{old})$
    $\Omega^{new} = RandomGeneration(\Omega^{old})$
    $ggmm_m = ggmm(f_T(M, \Omega^{new}))$
    $KLD^{new} = KLD_{MC}(ggmm_s || ggmm_m^{new})$
    if $KLD^{new} < KLD^{old}$ **then**
      $\Omega^{old} = \Omega^{new}$
    end if
  until some stopping criterion is satisfied
  Return $\Omega^* = \Omega^{old}$

optimization process can be further simplified, where R_α is the rotation matrix with an angle of α and t is a translation vector. The transformation of the data set is equivalent to the rigid change of the corresponding mixture model. We can obtain $\mu_k' = R_\alpha \mu_k + t$ and $\Sigma_k' = R_\alpha^T \Sigma_k R_\alpha$; the shape parameter β and scale parameter m are not affected during the transformation. In other words,

we only calculate the mixture models of the target scene and the scene to be registered through WMGGMM at the beginning, and the parameter estimation is not needed to be repeated during optimization. The transformation models used in the experiments below are all rigid transformations and all test data are generated based on the method in the previous section.

The unlikeness in sampling rate is a common challenge in point cloud registration. In general, the original scene will have a higher sampling rate to achieve more accurate modelling. In comparison, the scene to be registered may have a lower sampling rate due to the limitations of the sampling environment and tasks. Fig.2.5 shows the registration results of point sets with different sampling rates. The target scene (blue) has 1200 samples, while the scene to be registered (red) has 800. Although the sampling rate varies, the two scenes' data meet similar statistical distribution, so the method based on GGMM can still effectively register the two sets.

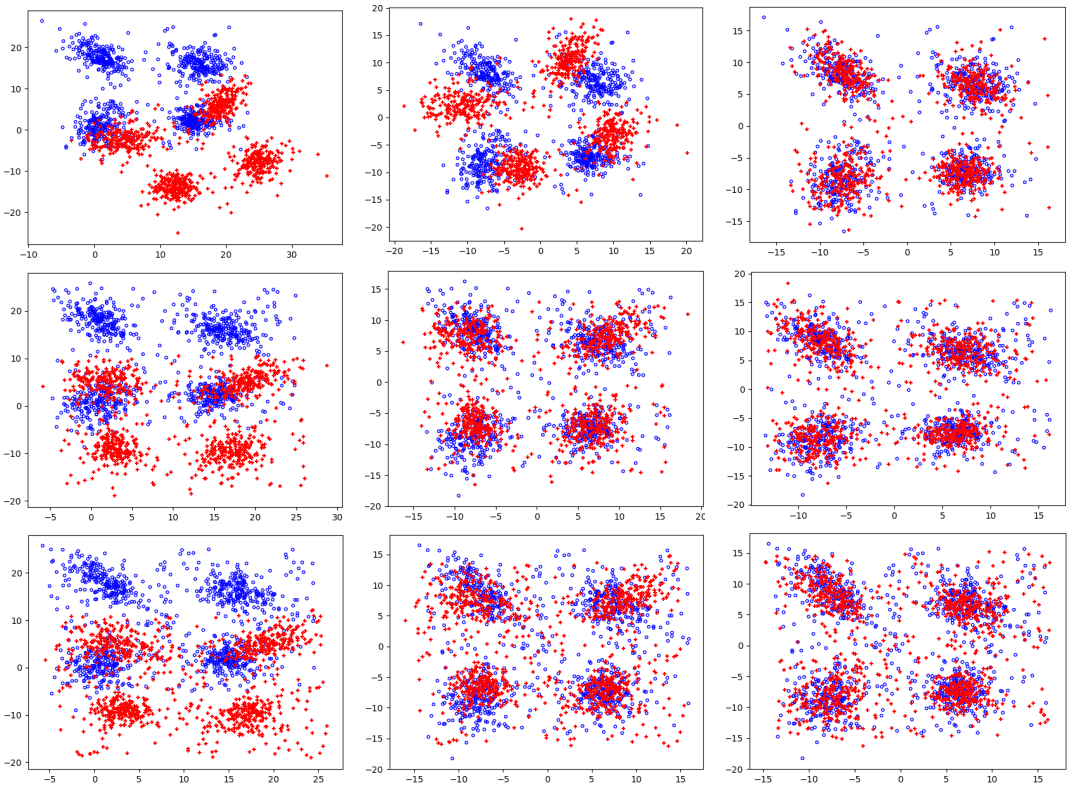


(a) original (b) centralized (c) registration

Figure 2.5: Registration result of point clouds with different sampling rates.

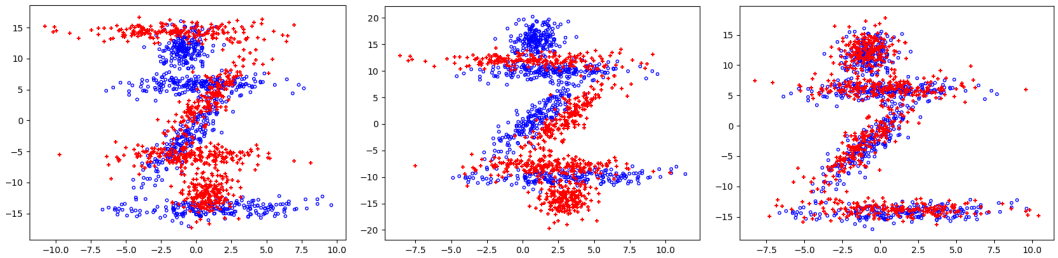
The registration results at different noise levels are given in Fig.2.6. The outcomes show that our method can effectively remove the noise and outliers' interference and extract the main features of the point cloud for matching, and the algorithm has good robustness.

In Fig.2.7, we designed a point cloud shaped like the Chinese character "zhi" and it is approximate center symmetrical. The distinction between it and the original graph after 180 degrees rotation is only at one "point." Hence, it is straightforward to fall into the optimal local solution in registration optimization. But, in several experiments, only 6% fell into local optima. Compared with derivative optimization, stochastic optimization provides the ability to jump out of the local optimal solution.



(a) original (b) centralized (c) registration

Figure 2.6: Registration results with 10% (top), 20% (middle), 30% (bottom) noise levels



(a) original (b) centralized (c) registration

Figure 2.7: The registration result when locally optimal solution exists.

2.4 Conclusion

This paper proposes a weighted multivariate generalized Gaussian mixture model and combines it with the stochastic optimization algorithm for point cloud rigid registration. This method requires enough samples in the registration scene to meet the minimum support of components. It extracts the data's mass features rather than the edge features (contour and shell), so it is suitable for substantial

point clouds with large data volume. The introduction of weights and the ability of generalized Gaussian to describe peak data can effectively reduce the influence of noise and outliers and obtain the critical features of data-intensive regions. Experimental results attest that the algorithm has sufficient robustness. The stochastic optimization algorithm reduces the coupling between algorithm modules, intensifies the algorithm's expansibility, and provides a more potent global optimization capability. However, in the mixture model's parameters estimation process, almost all parameters need to be learned iteratively; therefore, some performance is sacrificed to enhance the algorithm's accuracy. Future work will be committed to improve the parameters estimation approach to improve the algorithm's performance.

Chapter 3

Single Target Visual Tracking Using Color Compression and Spatially Weighted Generalized Gaussian Mixture Models

3.1 Introduction

Visual tracking is a process to find the state of an object of interest (such as position, size and rotation Angle) defined in the current frame in the subsequent frames of a video, which is a significant and challenging task within computer vision [96, 97]. It is a supplement of target detection since object recognition for each image will lead to massive computation and affect real-time performance. In contrast, target tracking reduces the system complexity by taking advantage of continuous information between frames. It has a wide range of applications in real life, such as surveillance [98, 99], self-driving vehicles [100], UAV tracking shooting [101–103], human-computer interaction (augmented reality) [104], robot navigation [105], smart transportation [106, 107], etc.

There are two basic assumptions in visual tracking: first, the spatial position and size of the

same object in a video should not change dramatically in two frames, which is inferred according to objective facts in the visual principle [108]; second, the target to be tracked and the background should have a certain degree of differentiation based on some features. For instance, frequency domain or texture features are more effective than colour features in tracking soldiers hidden in the jungle. Under the hypothesis above, the general framework of target tracking consists of four parts: (1) Target initialization. When having video sequence input, the target's state in the initial frame is automatically determined by manual marking or automatically via a detection algorithm. The target candidate box could have different forms according to different needs, such as rectangle, ellipse, skeleton, key points, contour, etc [109]. (2) Target description. This step includes the extraction of target features (colour, texture, and so on) and subsequent modelling. There are two main types of models (generative and discriminative), the latter considers both target and background, and it is more robust. (3) Target search. An optimization method is used to find the best matching candidate region according to the model. (4) Model update. Since the model can only describe the previous target information, it will be lost if it has a significant change in tracking. The difficulty of visual monitoring is to ensure the algorithm's accuracy, adaptability, and real-time performance while the target and environment constantly change. Appearance and scale changes caused by object movement or non-rigid deformation, shooting angle and ambient illumination variation, objects being blocked or beyond the field of view are likely to impact feature extraction, search strategy and the final tracking performance [47].

Many classical algorithms have been proposed to solve the above problems, such as distance measurement similarity [110–113], mean-shift [48,114,115], particle filtering [116,117], correlation filtering [53, 118], etc. With the continuous development of deep learning, combining traditional filtering methods with deep networks or only using neural network frameworks to design more robust trackers have aroused widespread interest [104], such as CREST [119], DLT [60], TCNN [120], FCNT [121], RTT [122], etc. Recently, several cutting-edge methods have been proposed. For example, Danelljan et al. built a tracker based on the probabilistic regression network trained through minimizing KL divergence, which can model label noise from inaccurate annotation and ambiguities in the task [123]. Chen et al. came up with SiamBAN, which equates the tracking problem with simultaneous classification and regression. This algorithm does not need to set a

priori box in advance and directly classifies the target and obtains the boundary by regression in a unified FCN [124]. In [125], the authors put forward the Siam R-CNN. They combined Siamese re-detection architecture and a dynamic programming algorithm. This method effectively dealt with the long-term occlusion problem using the initial information and the previous prediction results to distinguish the target and interference.

In this paper, our main contribution is to improve further the approach previously proposed in [5]. Each pixel in the candidate region is given a distance-based weight (pixels closer to the center of the area have higher importance) by applying the spatial kernel method in the reference work. The authors used weighted data in the elliptical region to build a Gaussian mixture model (GMM) to describe the target's feature distribution while unweighted data around the target region to construct the other GMM to present the background. The observation model employed to guide target updates merges object and environment information by removing similar components from the targe t model to the background model. The log-likelihood of pixels in the expected region is used as the loss function, and the gradient to the position directs target positioning. From another point of view, the logarithmic likelihood function acts as the weight in the mean-shift update. When updating the target size, they used a unique sampling method to solve the number of pixels coupling with the region's size. Simultaneously, backward tracking is adopted to judge whether the newly added components in the GMM belong to the target when the model is updated.

However, the colour feature distributions of the target and the background are not uniform and smooth in most cases. When the proportion of a specific colour is relatively high, the histogram will show an apparent sharp shape, which leads to the fact that GMM cannot describe the target feature well in some situations. Besides, simply removing components similar to each other in the target and background model results in the loss of some of the judgment knowledge since it does not consider the prior probabilities (mixing coefficients) of the components in different GMMs. Therefore, we apply the generalized Gaussian mixture model (GGMM), which has a more powerful peak data representation ability in our work. Concurrently, to hold other distribution types' compatibility, we use colour compression and reparameterization to modify the distribution of original characteristics to make it more adaptable to the GGMM and accelerate parameters convergence. Moreover, the logarithm of the ratio of each pixel's responsiveness to the target and background model indicates

the possibility that the pixel belonged to the potential target. The possibility value determined each pixel's segmentation weight based on the threshold, and the position and size of the target are updated according to this.

The paper is organized as follows. Section 3.2 will review the literature related to the algorithm proposed. Section 3.3 describes the algorithm's workflow in detail, including processing the original frames using colour compression and reparameterization, derivation of EM algorithm for the spatially weighted generalized Gaussian mixture model, establishing the target model and background model, and updating the position and size with segmentation thresholds. Section 3.4 will present experimental results and comparisons on public datasets. The conclusion is finally drawn in section 3.5.

3.2 Related Work

This section will introduce the literature associated with the critical techniques used in this article. It mainly involves the following three parts that have greatly inspired our work: colour compression and image segmentation, generalized Gaussian mixture model, hybrid model-based tracking.

3.2.1 Colour Compression and Image Segmentation

The significance of colour compression lies in removing image statistics and perceptual redundancy to improve image expression and storage efficiency since the human visual system has limitations on the resolution ability of colour signals with slight differences [126]. Its core task is to improve the compressed image's fidelity and make it as close as possible to the original image while maximizing the elimination of irrelevant information [127]. For example, JPEG is a well-known standard colour image compression technology [128]. However, computer understanding of image is based on the colour value vectors rather like human perception, so the colour compression in a broad sense can be interpreted as extract and retain the deciding colour characteristics of the image. It makes the machine still obtain adequate messages in treating the compressed image while reducing the amount of input data. In this case, colour compression is essentially image segmentation

based on clustering because areas with similar colours will be naturally connected to achieve image objects' semantic expression. For instance, the image compression and decompression methods proposed in [129] are appropriate for most dual-colour images by applying bit-map creation, run-length encoding and k-means clustering. A heuristic k-means algorithm based on flower pollination compresses medical images has been proposed in [130]. Within our work, colour compression with k-means is utilized for the preprocessing of image frames.

3.2.2 Generalized Gaussian Mixture Model

The generalized Gaussian distribution (GGD), proposed by Kelker in 1970 [131], belongs to the family of elliptic distribution [132]. Due to the shape parameter introduction, the generalized Gaussian has a comprehensive representation ability (from the peak distribution to the uniform distribution). The Gaussian and Laplace distributions are particular cases of GGDs. Therefore, it is widely used in signal and image processing [1]. Most parameter estimation techniques of multivariate GGD are based on the maximum likelihood estimation(MLE) or moment method [17, 84]. [1] resolved the scatter matrix's MLE based on fixed point equations by assuming that the shape parameter is within 0 and 1. The shape parameter hypothesis is moderate in most circumstances because uniform distribution with big beta is meaningless in most practical feature extraction applications, which implies that the data tends to subject to a non-informative distribution. In other words, the probability of data appearing in the feature space is almost the same. On account of the excellent property of GGDs, it is a natural choice for hybrid models. Multivariate Generalized Gaussian Mixture Model (MGGM) has many applications in various scenarios, such as defect detection [133], human motion recognition [28, 134, 135], blind source separation [136, 137], image denoising and segmentation [88, 138, 139], etc. This paper derives the EM algorithm of spatially weighted GGMM according to [1, 92]. Nevertheless, when using the Newton-Raphson method to estimate shape parameters, overflow error is likely to occur due to insufficient data supporting. Therefore, shape parameters are taken as the hyper-parameter that can be adjusted artificially in our contribution.

3.2.3 Hybrid Model-based Tracking

In target tracking, the mixture model (mainly GMM) can represent an object's appearance or assist the tracking procedure [5]. A dynamic Gaussian mixture model (DGMM) is offered for elliptical moving target tracking [140]. The Kalman filter estimates the actual values of model parameters in subsequent frames, which combines the past prediction with the current observation to achieve a more accurate model update. Changes to objects (add, merge, delete) are reflected in the GMM components, the idea in [5] is close to this. Another approach to applying the mixed model is shown in [141]. Meghana et al. used the background differential technique to achieve soccer players' trajectory tracking. They built GMM of background through the first few frames, then mask foreground to separate the athlete from the environment and provide more prominent target detection features. [142] presents an improved Gaussian mixture model that integrates dynamic pixel parameter updates, which solves the misjudgment and lag caused by too fast image frame changes. The application of GMM in [143] supports the tracking process in the surveillance video, that is, through GMM to precisely separate a specific person in the image into four regions from top to bottom. It facilitates the subsequent human face recognition (based on HOG and SVM methods) and tracking. Our paper employs GGMMs with different weighted rules for modelling the target and background simultaneously to compute pixels' responsiveness in the observation model.

3.3 Tracking Algorithm

The proposed algorithm details are specified in this section, including preprocessing, modelling of target and background with spatially weighted generalized Gaussian mixture models, and location and size updates using segmentation weights.

3.3.1 Preprocessing

The intention of accepting colour compression for preprocessing is as follows: (1) Assuming that the object's colour is single, uneven illumination or other factors may cause colour variation in saturation and lightness, which can be considered as constant colour with Gaussian noise added. Colour compression utilizes clustering to find the best substitution (centroid) of similar colours to

decrease noise to a certain extent. (2) Colour compression is likewise a process of image segmentation. Based on colour similarity, regions with the same semantics are naturally connected, while areas that cannot be merged will have noticeable distinctions from each other. Therefore, this potentially intensifies the contrast between the target and background. (3) From the histogram perspective, colour compression is equivalent to reducing the number of bars and increasing their width. It makes the probability density of pixels at each bar's median value gain, leading to leptokurtic colour distribution. This preprocessing further strengthens the importance of introducing the GGMM because it is suitable for peak distributions. Together, colour compression guarantees the distance between clusters and makes the components overlap of the GGMM smaller, improving parameter estimation accuracy, and accelerating the model convergence.

Since K-means is a well-known and straightforward technique, we will not add more details. After the clustering is complete, we will assign the centroid's value to all points in the corresponding group to perform colour compression. But it is worth noting that the selection of the number of clusters is critical. If the number is inadequate, colour compression will eliminate some colour features. It can be interpreted by the overperforming filter that removes not only noise but most of the valuable characteristics. Conversely, the calculation amount will increase if the number is too large, and the preprocessing cannot achieve the expected effect. The reasonable number of components will vary according to the colour richness in the video sequence. Generally, 10-20 clusters can be considered. Or we can apply other approaches to decide the optimal number automatically. Besides, the above operations will result in many pixels with an identical colour value. We notice that in the histogram, the centroid exists in isolation without other bars near it, which is likely to cause singularity when estimating the covariance matrix of the GGMM. Hence we added Gaussian noise with a small covariance (a diagonal matrix proportional to the number of pixels in the cluster) to each colour component to deal with this issue, which can be viewed as the reparameterization process exhibited in Fig.3.1. To show the outcome of preprocessing clearly and intuitively in Fig.3.2, we apply a small number of components, leading to the yellow line in the middle of the road being merged into other colour classes. From the visual effect aspect, the processed image's meaning has not been significantly changed compared with the original image. Still, the features in the corresponding colour histogram have been considerably modified. [htbp!]

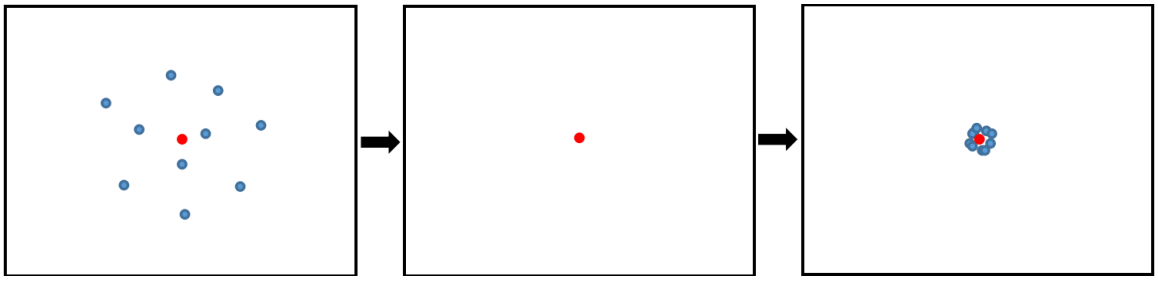


Figure 3.1: Reparameterization of pixels' colour distribution

Figure 3.2: Preprocessing modifications to color features

3.3.2 Spatially Weighted Generalized Gaussian Mixture Model

Spatial weighting is a kernel approach. The idea is that pixels in the target region do not contribute equally to the likelihood function because they have different importance according to their distance to the target center. A natural assumption is that pixels adjacent to the core are more likely to belong to the target [144]. To carry out distance weighting of pixels, we first necessitate acknowledging the target region's appearance representation. Here, we adopt the same approach (ellipse) as in [5]. We believe that the ellipse specifying the target is known in the head frame. A manually labelled rectangular box determines the center, short and long axes of it. In the subsequent tracking process, the current frame ellipse's initial parameters are the same as the immediately previous

frame parameters. The ellipse expressing the target is defined as follows:

$$f\left(\mathbf{x}_n;\mathbf{y},\mathbf{h}\right)=\left(\mathbf{x}_n-\mathbf{y}\right)^T\mathbf{H}^{-1}\left(\mathbf{x}_n-\mathbf{y}\right)=\left[\frac{x_{n1}-y_1}{h_1}\right]^2+\left[\frac{x_{n2}-y_2}{h_2}\right]^2 \tag{38}$$

where $\mathbf{x}_n=[x_{n1}\ x_{n2}]^T$ is the position of the n^{th} pixel in the frame, $\mathbf{y}$ is the center coordinates of the ellipse, $\mathbf{h}=[h_1\ h_2]^T$ represents the long and short axis of the ellipse and $\mathbf{H}=diag(h_1^2,h_2^2)$. We notice that this is the Mahalanobis distance squared between $\mathbf{x}_n$ and $\mathbf{y}$. Simultaneously, this expression normalizes the pixels inside the ellipse, making their relative position value within [-1, 1]. As mentioned earlier, the weight of each pixel depends on how far it is from the center, so we have:

$$w_n(\mathbf{y})=k[f(\mathbf{x}_n;\mathbf{y},\mathbf{h})] \tag{39}$$

where k denotes the kernel function, which should be degressive for the pixels inside the ellipse, and it makes the rest of pixels have zero weight. The selection of kernel function is not unique, as long as the above properties are satisfied. However, some kernel functions can not guarantee that $g(x)$ defined in section 3.3.3 is still a decreasing function, which we will further explain in the content of the position update. We choose the kernel function as following (σ is the bandwidth with default setting as 1):

$$k(x)=\begin{cases} e^{-x/\sigma} & (x\leq 1) \\ 0 & (\text{otherwise}) \end{cases} \tag{40}$$

The preceding content illustrates how the weights of pixels are constructed. The next step is to introduce the consequences into the GGMM. Here we use the probability density function of MGGD defined in [1]:

$$N_g(I;\mu,\Sigma,\beta)=\frac{\Gamma(d/2)\beta}{\Gamma(d/2\beta)\pi^{d/2}2^{d/2\beta}|\Sigma|^{1/2}}e^{-\frac{1}{2}\left[(I-\mu)^T\Sigma^{-1}(I-\mu)\right]^\beta} \tag{41}$$

where $\Gamma(\cdot)$ denotes the Gamma function, $I,\mu\in R^d$ and μ is the mean vector, is d×d real symmetric positive definite matrix called scatter matrix; $\beta>0$ is the shape parameter of MGGD. Thus we can

acquire the log-likelihood function of the n^{th} pixel:

$$L_n = \ln P\left(I_n; \boldsymbol{\theta}\right) = \ln \sum_{k=1}^{K} \pi_k N_g\left(I_n; \mu_k, \Sigma_k, \beta_k\right) \tag{42}$$

where I_n is the feature vector corresponding to the n^{th} pixel, which is three-dimensional in the color RGB image. The GGMM in above equation has k components and the mixing coefficient is π_k which satisfies $\sum_{k=1}^{K} \pi_k = 1$. Accordingly, the weighted likelihood in the elliptic region is given by:

$$L(\mathbf{I}, \mathbf{w}(\mathbf{y}), \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta}) = \sum_{n=1}^{N} w_n(\mathbf{y}) L_n = \sum_{n=1}^{N} w_n(\mathbf{y}) \ln \sum_{k=1}^{K} \pi_k N_g\left(I_n; \mu_k, \Sigma_k, \beta_k\right) \tag{43}$$

where N is the number of pixels, $\mathbf{I} = \{I_n\}_{n=1}^{N}$, $\mathbf{w}(y) = \{w_n(\mathbf{y})\}_{n=1}^{N}$, $\boldsymbol{\Theta} = \{\pi, \mu, \boldsymbol{\Sigma}, \beta\} = \{\theta_k\}_{k=1}^{K}$ and $\theta_k = \{\pi_k, \mu_K, \Sigma_k, \beta_k\}$, it is worth noting that the weights here are not normalized. To renew the model parameters using the EM algorithm, we introduce the corresponding hidden variable $z_n = [z_{n1}, \ldots, z_{nk}]$ for each pixel. The log-likelihood of the complete data constituted by $\{\mathbf{I}, \mathbf{Z}\}$ is:

$$L = \ln p(\mathbf{I}, \mathbf{w}(\mathbf{y}), \mathbf{Z}; \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{44}$$

The posterior probability is proportional to the joint probability density according to the Bayesian formula since the hidden variables are unknown. Then we obtain:

$$L = \ln p(\mathbf{Z}; \mathbf{I}, \mathbf{w}(\mathbf{y}), \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \propto \prod_{n=1}^{N} \prod_{k=1}^{K} \left[\pi_k N_g\left(I_n; \mu_k, \Sigma_k, \beta_k\right)\right]^{z_{\mathrm{nk}} \mathrm{w_n}(\mathbf{y})} \tag{45}$$

The disagreement with the standard GMM definition is that each observation exists with probability $w_n(\mathbf{y})$ instead of one. The $\gamma_{nk} = E[z_{nk}]$ can be obtained from this so that we get the Q function:

$$Q_\theta = \sum_{n=1}^{N} w_n(\mathbf{y}) \sum_{k=1}^{K} \gamma_{nk} \left[\ln \pi_k + \ln N_g\left(I_n; \mu_k, \Sigma_k, \beta_k\right)\right] \tag{46}$$

$$\gamma_{nk} = \frac{\pi_k N_g\left(I_n; \mu_k, \Sigma_k, \beta_k\right)}{\sum_{k=1}^{K} \pi_k N_g\left(I_n; \mu_k, \Sigma_k, \beta_k\right)} \tag{47}$$

Through taking the gradient of Q with respect to π, μ and Σ, then setting derivatives equal to 0, we get the following update form to ensure that the log-likelihood is maximized:

$$\pi_k = \frac{\sum_{n=1}^{N} w_n(\mathbf{y})\gamma_{nk}}{\sum_{n=1}^{N} w_n(\mathbf{y})} \tag{48}$$

$$\mu_k = \frac{\sum_{n=1}^{N} w_n(\mathbf{y})\gamma_{nk}\left[(I_n-\mu_k)^T\Sigma_k^{-1}(I_n-\mu_k)\right]^{\beta_k-1} I_n}{\sum_{n=1}^{N} w_n(\mathbf{y})\gamma_{nk}\left[(I_n-\mu_k)^T\Sigma_k^{-1}(I_n-\mu_k)\right]^{\beta_k-1}} \tag{49}$$

$$\Sigma_k = \frac{\beta_k}{\sum_{n=1}^{N} w_n(\mathbf{y})\gamma_{nk}} \sum_{n=1}^{N} \frac{w_n(\mathbf{y})\gamma_{nk}(I_n-\mu_k)(I_n-\mu_k)^T}{\left[(I_n-\mu_k)^T\Sigma_k^{-1}(I_n-\mu_k)\right]^{1-\beta_k}} \tag{50}$$

It should be noted that μ_k and Σ_k do not have closed forms, and their updated formulas are fixed-point equations, which require several iterations to achieve the final result. Paper [1] proves that the solutions exist and converge under the condition when β is within 0 and 1. We use the Frobenius norm to measure the difference between θ_k^t and θ_k^{t+1} and the loop stop condition satisfies $||\Delta\theta_k||_F \leq \varepsilon$. Instead of estimating the mixed model's shape parameter, we treat it as a tunable hyperparameter (the default value is 0.8 and all β_k have the same setting) to improve the hybrid model's representation ability for the peak distribution. The principal reason is that in parameters updating, the valid point is only in the ellipse. The insufficient number of observed samples will lead to inadequate component support when updating β.

We apply the MML criterion as the cost function and terminate refreshing the entire mixed model's parameters when it changes slightly [92]. The components are updated one by one, and those that do not meet the threshold will be deleted during the updating process as well as the mixing coefficient will be reassigned:

$$\boldsymbol{\theta}_{MML} = \arg\min_{\theta}\left\{\frac{M}{2}\sum_{k\in\mathbf{K}^+}\ln\pi_k - Q_\theta + \frac{K^+(M+1)}{2}\left[1+\ln\frac{N}{12}\right]\right\} \tag{51}$$

where $M = d(d+3)/2$ is the number of free parameters in each component (in the case of using a positive definite symmetric covariance matrix) and d is the dimensions of color features. $\mathbf{K}^+$ is the set of non-empty components and K^+ is the number of elements in it. The calculation formula for

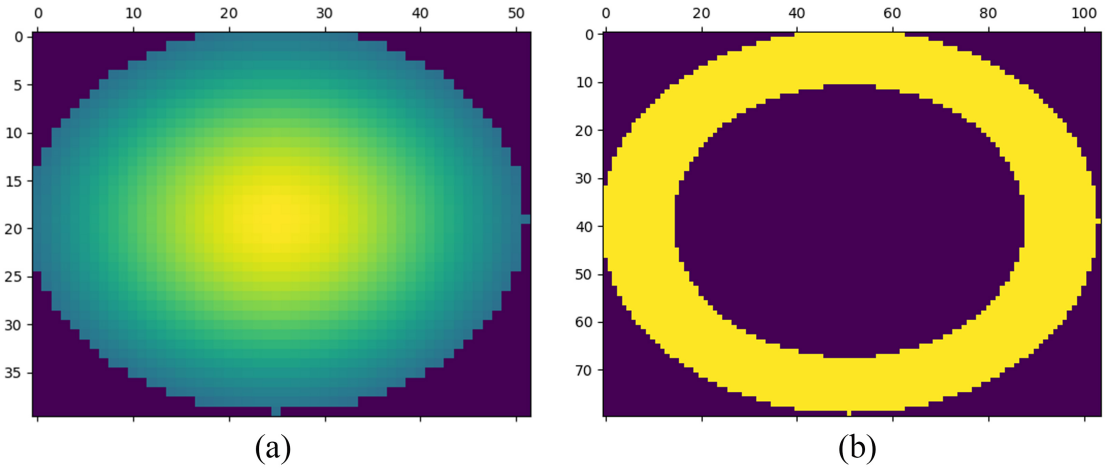


Figure 3.3: Weight masks for the target and background

π_k can be modified as:

$$\pi_k = \frac{\max\left(0, \sum_{n=1}^{N} w_n(\mathbf{y})\gamma_{nk} - MK^+/2\right)}{\sum_{k=1}^{K} \max\left(0, \sum_{n=1}^{N} w_n(\mathbf{y})\gamma_{nk} - MK^+/2\right)} \tag{52}$$

When modelling with GGMM, only the pixels inside the ellipse participate in the updating (pixels with a weight of 0 outside the ellipse have no contribution). Therefore we use the ellipse's outer rectangle to form the ROI (region of interest) instead of the whole frame to withdraw the unnecessary computational overhead. To overcome background interference to the tracking process, we select the annular area around the target to build the background model, i.e. it is twice the object ellipse and does not contain the target region. There are different methods when picking weighting for the background model, such as treating each pixel equally or giving more weight to pixels farther away from the center (which are more likely to belong to the background). We consider the former in this article, and Fig.3.3 shows the weight masks for the target and background (note the size diversity shown on the axes). The fundamental steps for describing the object and context using GGMM are presented in Algorithm 3.

Algorithm 3 Spatially Weighted Generalized Gaussian Mixture Models

Require: K_{max}, $\mathbf{I} = \{I_n\}_{n=1}^{N}$, $W = \{w_n\}_{n=1}^{N}$,
$\Theta^{(0)} = \{\pi_k^{(0)}, \mu_k^{(0)}, \Sigma_k^{(0)}\}_{k=1}^{K_{max}}$, $\beta = \{\beta_k\}_{k=1}^{K_{max}}$
Ensure: Optimal mixture model parameters Θ^*
Set: $r = 0$, $MML^{(0)} = +\infty$
repeat
for $k = 1$ to K_{max} **do**
// E - Step:
$\gamma_{nk}^{(r+1)} = \frac{\pi_k^{(r)} N_g(I_n; \mu_k^{(r)}, \Sigma_k^{(r)}, \beta_k^{(r)})}{\sum_{l=1}^{K_{max}} \pi_l^{(r)} N_g(I_n; \mu_l^{(r)}, \Sigma_l^{(r)}, \beta_l^{(r)})}$
Compute the # of non-empty components:K^+
// M - Step:
$\pi_k^{(r+1)} = \frac{max(0, \gamma_{nk}^{(r+1)} w_n - MK^+/2)}{\sum_{l=1}^{K_{max}} max(0, \gamma_{nl}^{(r+1)} w_n - MK^+/2)}$
if $\pi_k^{(r+1)} > 0$ **then**
// Update μ_k:
repeat
$\mu_k^{new} = FP_\mu(\mu_k^{old})$
until $\|\Delta\mu_k\|_F < \epsilon$
$\mu_k^{(r+1)} = \mu_k^{new}$
// Update Σ_k:
repeat
$\Sigma_k^{new} = FP_\Sigma(\Sigma_k^{old})$
until $\|\Delta\Sigma_k\|_F < \epsilon$
$\Sigma_k^{(r+1)} = \Sigma_k^{new}$
end if
end for
Compute $MML^{(r+1)}$
$r = r + 1$
until $|\Delta MML| < \epsilon$
return Θ of non-empty components as Θ^*

3.3.3 Location Update and Segmentation Weights

Assuming that we only consider the target model for the moment, position updating is essentially looking for the location where the likelihood function of the pixels in the elliptical region is maximized. To reach the target's position in the next frame, we need to figure the gradient of the log-likelihood for position $\mathbf{y}$:

$$\frac{dL}{d\mathbf{y}} = \frac{dL(\mathbf{I}, \mathbf{w}(\mathbf{y}), \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\beta})}{d\mathbf{y}} = \sum_{n=1}^{N} \frac{d\mathrm{k}\left(f\left(\mathbf{x}_n; \mathbf{y}, \mathbf{h}\right)\right)}{d\mathbf{y}} L_n \tag{53}$$

By setting the above equation equal to zero and defining the negative derivative of the kernel function as $g(x) = -dk(x)/dx$, we can obtain the update formula for $\mathbf{y}$ in a mean-shift way, and it will stop when the change in position is less than one pixel. As suggested earlier, although there are multiple kinds of kernel functions satisfying monotonous decrease in the interval $[0, +\infty)$, the $g(x)$ gained from it has various monotonicity. As a result, pixels in the mean-shift update have different spatial weighting methods according to the choice of $k(x)$. For example, when $k(x) = -x^2 + 1$, $g(x) = 2x$ (the weight increases with distance); When $k(x)$ is equal to $-x + 1$, $g(x) = 1$ (the weight is fixed independent of the distance). The kernel function we choose ensures the consistency of the spatial weighting approach, i.e. $k(x) = e^{-x} = g(x)$.

$$\mathbf{y}_{new} = \frac{\sum_{n=1}^{N} \mathbf{x}_n g\left(f\left(\mathbf{x}_n; \mathbf{y}_{old}, \mathbf{h}\right)\right) L_n}{\sum_{n=1}^{N} g\left(f\left(\mathbf{x}_n; \mathbf{y}_{\text{old}}, \mathbf{h}\right)\right) L_n} \tag{54}$$

Besides, we can see that L_n acts as the information weight in the position update formula. In other words, it denotes the role of pixel colour responsiveness of the model in the update. However, the probability is generally less than 1 when bringing a pixel sample into the GGMM (its logarithm is negative), which may cause the position to move outside the ellipse. Also, the current L_n does not reflect the impact of the background. Hence we make the following definition then get the modified L_n:

$$\mathrm{r_n} = L_t\left(I_n\right) - L_b\left(I_n\right) = \ln\left[P_t\left(I_n\right)/P_b\left(I_n\right)\right] \tag{55}$$

$$L_n = \begin{cases} 1 & \text{if } L_t\left(I_n\right) \geq \delta \text{ and } r_n \geq \tau \\ 0 & \text{otherwise} \end{cases} \tag{56}$$

Where $L_t(I_n)$ and $L_b(I_n)$ represent the logarithm of the responsiveness of the n^{th} pixel to the target and background models, respectively. δ (taking -10 as default) and τ (pick from 0 to 0.4) are the thresholds.

Fig.3.4 exhibits several conditions in which the pixel responds to the target and background model. A one-dimensional diagram is used for the convenience of presentation. The horizontal axis denotes the colour feature space, and the vertical axis signifies the probability (responsiveness). The red and blue curves are the target and background GGMMs (it is simplified here by the single peak

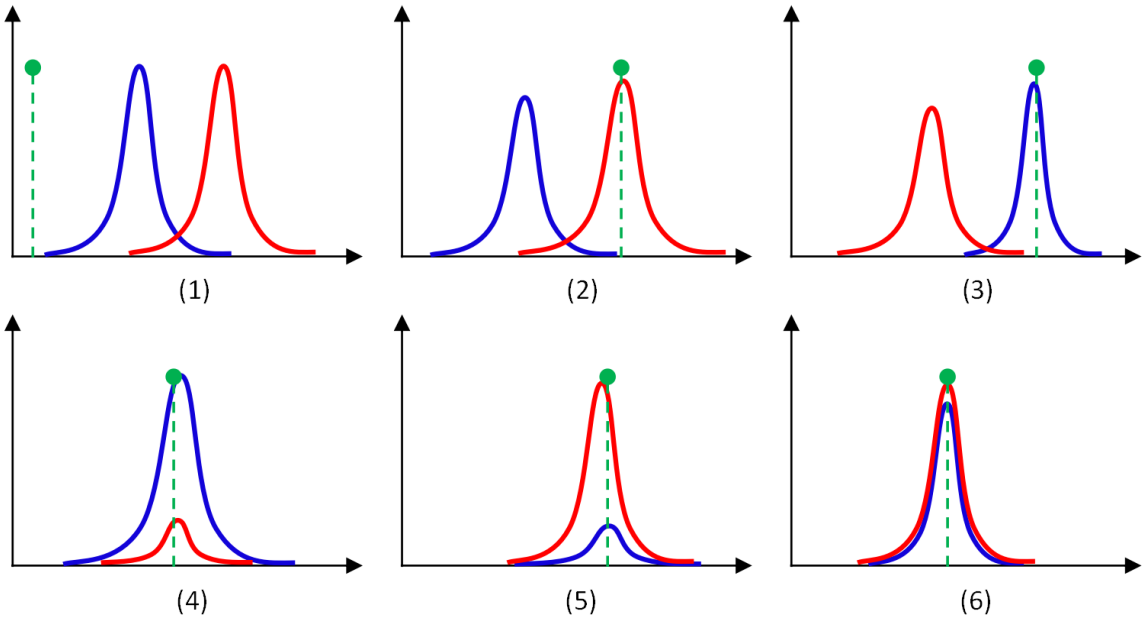


Figure 3.4: Various examples of pixel responsiveness to target and background models

distributions), and the green dotted line expresses the pixel feature value. When the responsiveness of these two GGMMs approaches zero (the first one in Fig.3.4), it means that the current pixel colour scarcely exists in the target and background modelling region. Although it is unlikely to judge whether it pertains to the unmodeled background area, it does not belong to the target. When the object or background's responsiveness is close to 0 while the environment or target's responsiveness is powerful, it is obvious to conclude the pixel relates to which side, as displayed by cases 2 and 3 in Fig.3.4.

When the target and background responsiveness are relatively high simultaneously, the two GGMMs have similar components with different mixing coefficients, which indicates the target and the background have the same colour zone, but their proportion is distinctive. If the gap between them is significant, it can be explained as the target region contains a small portion of the back scene. Subplots 4 and 5 of Fig.3.4 show this from the perspective of the pixel belonging to the background and target, respectively. If there is a slight discrepancy between the two, as in situation 6 of Fig.3.4, we need to introduce a threshold to decide. Only when the target's responsiveness is greater than that of the background to some extent, the pixel can be considered an element of the target. However, such a comparable situation may also occur in case 1, so we need to consider the ratio and the

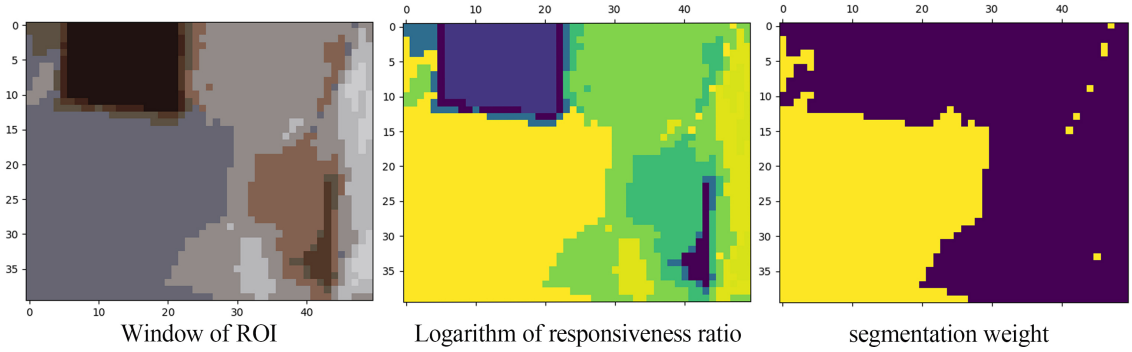


Figure 3.5: The segmentation weights of the pixels

absolute value concurrently. The pixel is directly determined to belong to the background if the target responsiveness is lower than the threshold delta. The summary of the above situation revises L_n. The background and target are separated by this method, so L_n is also called segmentation weight, and Fig.3.5 reveals this process. In addition, if the target moves too quickly, it is likely to disappear in the ROI. At this time, almost all segmentation weights will be 0 and position updates will not be able to continue (the denominator in the mean-shift formula approaches 0). Even if the target is at the edge of the ROI, these pixels with segmentation weights of 1 contribute very little due to the distance weighting of the kernel function, leading to the position change is less than one pixel at the beginning. Since the sum of the segmentation weights expresses the size of the target area, when it is less than 1/10 of the number of pixels in the ROI, we assume that the target does not exist in the current ROI. Still, we cannot determine whether the object disappears in the video frame. We adopt the method below to produce a rough prediction for the next trajectory point (position) of the target and use the predicted value to reselect ROI for the mean-shift update. If the sum of weights still has a similar situation as above, the tracking is considered to have failed. In reality, the target's movement is continuous and smooth (for example, the object cannot suddenly turn at an acute angle). The inaccuracy of the position is from the low sampling rate. Fig.3.6 reveals the process of location prediction based on this. The most recent four target positions $\mathbf{y}^{(t-3)}, \ldots, \mathbf{y}^{(t)}$ are stored in tracking. Accordingly, we can further obtain the velocity and acceleration vectors of the target as $\mathbf{v}^{(t)} = \mathbf{y}^{(t)} - \mathbf{y}^{(t-1)}$ and $\mathbf{a}^{(t)} = \mathbf{y}^{(t)} - 2\mathbf{y}^{(t-1)} + \mathbf{y}^{(t-2)}$, respectively. After applying the weighted average, the predicted acceleration vector is $\mathbf{a}^{(t+1)} = \alpha\mathbf{a}^{(t)} + (1-\alpha)\mathbf{a}^{(t-1)}$, where

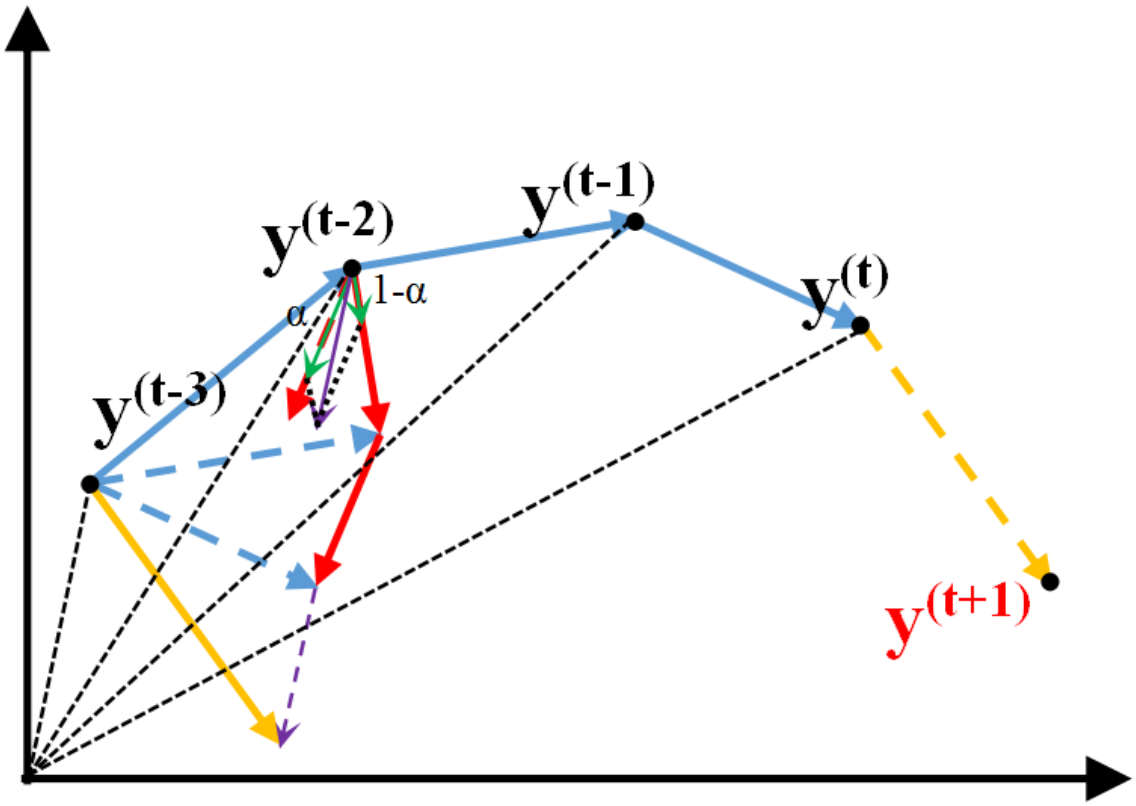


Figure 3.6: The process of location prediction when the target disappears from the ROI

$\alpha \in (0, 1)$, and α is 0.7 in our case. Then the predicted position is represented as:

$$\begin{aligned} y^{(t+1)} &= y^{(t)} + v^{(t+1)} = y^{(t)} + v^{(t)} + a^{(t+1)} \\ &= (\alpha + 2)y^{(t)} - 3\alpha y^{(t-1)} + (\alpha - 2)y^{(t-2)} + (1 - \alpha)y^{(t-3)} \end{aligned} \tag{57}$$

3.3.4 Scale Adaptation

The reference [5] designates that the method based on gradient updating is not feasible because its derivative is always negative, and h will converge to only one pixel size finally. Since pixels' number in ellipse changes with its size, and the increase of the likelihood function is proportional to the decrease of the size of the ellipse, the method based on scaling the ellipse also fails to work. To solve the preceding problems, we still apply the segmentation weights to estimate size parameters. Intuitively, the optimal size of the ellipse can be obtained when the summation of importance for all pixels in the ellipse is maximum. But since the weights of the background region are 0, virtually

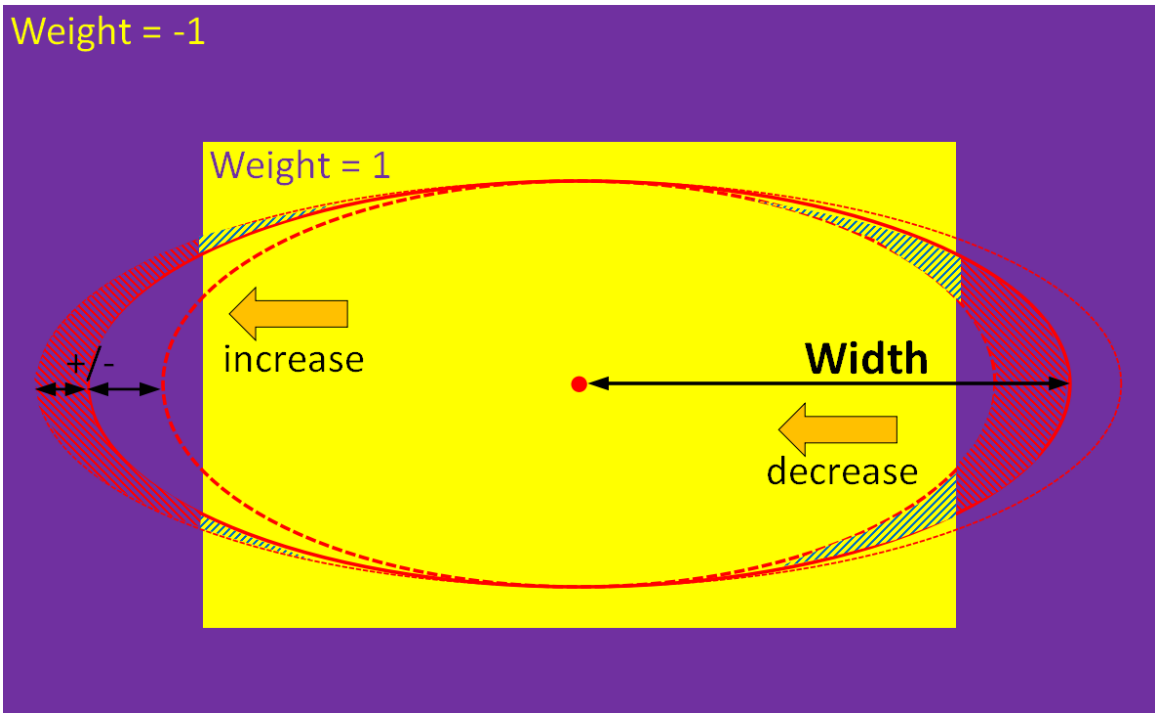


Figure 3.7: The principle of size adaptation

any ellipse containing the target will have the same weights aggregation. By making the weights as -1, we introduce a penalty mechanism for the background pixels. Fig.3.7 considers an idealized scenario to demonstrate the principle of this method. Considering the parameters of the major and minor axes of the ellipse are alternately and independently determined in the same way until they converge, we take the horizontal direction as an example. The target area is a yellow rectangle, and the initial ellipse in the present frame is the solid red line. The long axis of the ellipse increases will lead to the background area (red shade) gains more than the target area (green shade), resulting in a negative increment of the weights-sum. When the horizontal size of the ellipse decreases, the reduction of the background region is more significant than that of the target region, so the weights-sum raises. According to the optimization goal of making the weight-sum maximum, the horizontal size of the ellipse should move to the direction of decline. When the ellipse is exactly tangent to the target, all the pixels in the ellipse belong to the target. If the horizontal size continues to decrease, then the weights-sum will not be at the maximum. Therefore, the ellipse eventually becomes an inscribed oval of the target in this case. This approach will make some tradeoffs between the target and background area when discovering size parameters because the shape of most objects is not

necessarily regular in practice.

We scale up or down the size of two pixels every time for heuristic optimization in experiments, based on the assumption that the change of the target between successive frames is usually tiny. The same idea also can determine the rotation angle of the target. With a fixed size ellipse, we try to rotate it by 1 or -1 degree at a time to find the angle that maximizes the weights-sum. But for most papers, the tracking problem focuses on the object's trajectory rather than the attitude change. The rotation parameters are ignored in this paper to facilitate the comparison of results with other methods. The progress of size updating is given in Algorithm 4 .

Algorithm 4 Ellipse Size Updating Algorithm

Require: Color compressed frame **F**, location **y**,
initial size $\mathbf{h} = [h_1, h_2]$, target model M_t, background model M_b
Ensure: Optimal size parameters $\mathbf{h}^*$
$cur\_roi = get\_roi(\mathbf{F}, \mathbf{y}, \mathbf{h})$
$w\_sum^* = get\_weightsum(cur\_roi, \mathbf{h}, M_t, M_b)$
$\mathbf{h}^* = \mathbf{h}$
repeat
while True do
$h_1 = h_1^* + 2$
$cur\_roi = get\_roi(\mathbf{F}, \mathbf{y}, \mathbf{h})$
$cur\_w\_sum = Get\_weightsum(cur\_roi, \mathbf{h}, M_t, M_b)$
if $cur\_w\_sum \leq w\_sum^*$ **then**
break
else
$w\_sum^* = cur\_w\_sum$
$h_1^* = h_1$
end if
end while
same loop as above with $h_1 = h_1^* - 2$
same loop as above with $h_2 = h_2^* + 2$
same loop as above with $h_2 = h_2^* - 2$
until $|\Delta w\_sum^*| < \epsilon$
return $\mathbf{h}^*$ as optimal size parameters

3.3.5 Model Update

In the tracking process, the target and background feature distribution is constantly changing. The colour addition and removal caused by the rotation of cubes with different colours on each surface in the three-dimensional space is a typical example. So the corresponding model needs to be

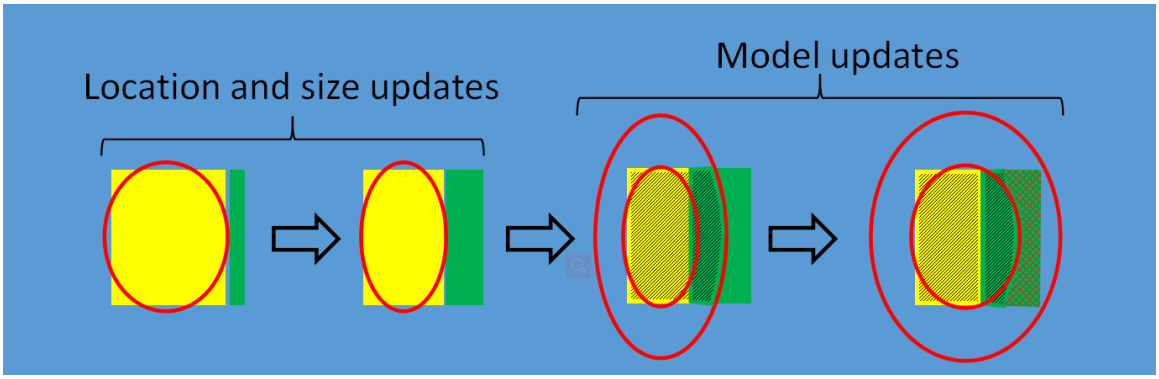


Figure 3.8: Model update diagram

dynamically updated to ensure accuracy. The reconstruction of the model is based on the selection determined by the algorithm in the previous frame. Still, this result does not necessarily represent the actual position and size of the target, and manual correction would render the tracking of the target meaningless. Therefore, new mechanisms need to be introduced to deal with this problem.

Under the situation mentioned above, the target's newly increased colour should have evident differences from the colours that exist in the object and background region. Otherwise, the area with this colour will be assigned to the target or background based on current models when calculating segmentation weights. The GGMM will not create new components during the updating. According to the physical continuity of the target, the newly added colour area is in the neighbourhood of the target, that is, within the background ellipse. Therefore, we prioritize the rebuilding of the GGMM for the background region and use $M_b^{(t+T)} = \{C_i^{(t+T)}\}_{i=1}^{m}$ to represent it. The previous background model is $M_b^{(t)} = \{C_j^{(t)}\}_{j=1}^{n}$, where C denoted the center of the model component. We obtain the following formula by applying the Euclidean distance :

$$d_j = \min \left\{ \left\| C_j^{(t)} - C_i^{(t+T)} \right\|^2 \right\}_{i=1}^{n} \tag{58}$$

where d_j measures the distance between the two closest components in $M_b^{(t+T)}$ and $M_b^{(t)}$. When it is larger than the threshold introduced to deal with colour drift, $C_j^{(t+T)}$ is considered the newly added colour. If not, it means the component already exists in the previous background model.

Based on the rate of change of the ellipse aspect ratio, we can determine whether the new component belongs to the target or the background because panning and zooming of the target

during tracking do not change the ellipse's aspect ratio, but rotation does. So when $|h_1^{(t)}/h_2^{(t)} - h_1^{(t+T)}/h_2^{(t+T)}|/|h_1^{(t)}/h_2^{(t)}| > \xi$, we consider that the added colour belongs to the target and ξ is a self-defined threshold that determines the sensitivity of model updates. The segmentation weights

Algorithm 5 Weighted Generalized Gaussian Mixture Models Based Tracking Algorithm

Require: Video sequence $\mathbf{S} = \{\mathbf{F}_i\}_{i=1}^n$, model update interval T,
the initial parameters Θ_t and Θ_b of GGMMs, the number of compressed colors k
Ensure: The position and size of the ellipse in each frame
for i=1 **To** n **do**
resize $\mathbf{F}_i$ to the appropriate resolution
if i==1 **then**
initialize **y** and $\mathbf{h} = [h_1, h_2]$ by manual selection
$\mathbf{F}_i = colour\_compress(\mathbf{F}_i, k)$ with Gaussian noise
$target\_roi = get\_roi(\mathbf{F}_i, \mathbf{y}, \mathbf{h})$
calculate the target space weights $\mathbf{W}_t$
generate the target model $M_t = GGMM(\mathbf{F}_i, \mathbf{W}_t, \Theta_t)$
$background\_roi = get\_roi(\mathbf{F}_i, \mathbf{y}, \mathbf{h} * 2)$
calculate the background space weights $\mathbf{W}_b$
generate the background model $M_b = GGMM(\mathbf{F}_i, \mathbf{W}_b, \Theta_b)$
else
$\mathbf{F}_i = colour\_compress(\mathbf{F}_i, k)$
while $||\Delta\mathbf{h}||^2 > 0$ pixel **do**
$cur\_roi = get\_roi(\mathbf{F}_i, \mathbf{y}, \mathbf{h})$
compute the segmentation weights $\mathbf{W}_s = seg\_weights(cur\_roi, M_t, M_b)$
if $\mathbf{W}_s < 0.1*$ pixels number in roi **then**
roughly predict the location and update the roi and $\mathbf{W}_s$
else
stop tracking
end if
$\mathbf{y} = pos\_update(\mathbf{y}, cur\_roi, \mathbf{W}_s)$
end while
using algorithm 2 to update the ellipse size **h**
end if
if i **mod** T == 0 **then**
M_t and M_b are updated according to the ellipse aspect ratio changes
end if
end for

of the area with this colour in the background ellipse will be corrected to one. Then we will use the previous methods to adjust the position and size of the target ellipse to cover the changed target area and rebuild the target model, which will spontaneously eliminate the components of the disappeared colour. Simultaneously, we should delete the corresponding parts in the background model, and the

mixing coefficient of other members should be normalized again. When the new colour belongs to the background, we update the target model directly. The interval of the model update is fixed, and it is refreshed every T frames. The process of updating the model is illustrated in Fig.3.8. The overall procedure of the initialization and tracking is displayed in the weighted generalized Gaussian mixture models based tracking (WGGMMT) Algorithm 5.

3.4 Experimental Results

We use seven real datasets to evaluate the proposed algorithm. These video sequences are selected from the VOT2020 database, including book, girl, hand2, helicopter, polo, road and wheel. Each series contains 50 frames without complete occlusion on the target. To reduce the time overhead of colour compression, we first resize the video frame. Therefore, the original annotations provided in the dataset are no longer available. We determine the target ground truth (both position and size) in the video sequence using the average results of multiple manual labelling. It should be noted that both [5] and our algorithm use ellipse to represent the target. To facilitate the comparison with the ground truth, we utilize the long and short axes of the ellipse to obtain the bounding rectangle of the ellipse to carry out the relevant calculation.

There are three approaches used to compare with our method. We have reproduced the WLT and WLTMS algorithms proposed in the primary references [5]. For the camshift algorithm, we use the OpenCV library to implement. All parameter settings in the above comparison are consistent with those in [5]. We employ the same six assessment criteria as in [5] and [144]: (1) The number of frames tracked correctly, which is a rough measure as long as the outer rectangle of the ellipse covers more than 25% of the actual area. (2) The position error is obtained from the Euclidean distance between the estimated center and actual position dividing by the diagonal of the ground truth rectangle. (3) The Euclidean distance between the estimated size vector $\mathbf{h} = [h_1\ h_2]$ and the actual size vector normalized by the diagonal of the ground truth rectangle specifies the size error. The above two error computation methods eliminate the influence caused by the size difference of

the target. (4) The average precision defined as:

$$P = \frac{1}{N}\sum_{n=1}^{N} p_n \tag{59}$$

$$p_n = \frac{\#\text{ of correctly tracked pixels in frame n}}{\#\text{ of tracked pixels in frame n}} \tag{60}$$

(5) The average recall defined as:

$$R = \frac{1}{N}\sum_{n=1}^{N} r_n \tag{61}$$

$$r_n = \frac{\#\text{ of correctly tracked pixels in frame n}}{\#\text{ of ground truth pixels in frame n}} \tag{62}$$

(6) The average F-measure defined as:

$$F = \frac{2}{N}\sum_{n=1}^{N} \frac{p_n \times r_n}{p_n + r_n} \tag{63}$$

Tables 3.1 - 3.6 and Figures 3.9 - 3.15 display the quantitative comparison results of these four methods. Experiments show that no single approach can be applied to all video sequences, and each method has its advantages and disadvantages. On the whole, our WGGMMT algorithm can achieve a similar capability to the compared ones. The average performance of tracking position error is better than that of other methods in the majority of cases because of the introduction of target and background responsiveness ratio. However, in terms of the average size error, the test outcomes reveal a certain gap between our method and other techniques caused by our size update rule based on the sum of segmentation weights. The non-rigid deformation of the object will result in the smallest rectangular box containing the aim region mingled with a large number of background pixels. The penalty brought by them will make the ellipse representing the target tend to be more diminutive, leading to the target area is the complete subset of the ground truth most time. Therefore, the reflection on performance is preferable precision but lower recall.

In video sequence one, the camera's view is almost fixed, so the background does not change significantly. However, the book has drastic and quick movements, including translation, turning and scaling, which means that the change between frames is conspicuous. The camshift and WLTMS

Table 3.1: Performance of camshift, WLT, WLTMS and WGGMMT in terms of correct target localization

Seq.	camshift	WLT	WLTMS	WGGMMT
Book	19/50	50/50	22/50	50/50
Girl	50/50	50/50	50/50	50/50
Hand2	50/50	50/50	50/50	50/50
Helicopter	50/50	50/50	50/50	50/50
Polo	50/50	50/50	49/50	50/50
Road	49/50	50/50	50/50	50/50
Wheel	46/50	50/50	48/50	50/50

Table 3.2: Performance of camshift, WLT, WLTMS and WGGMMT in terms of position error (mean±std)

Seq.	camshift	WLT	WLTMS	WGGMMT
Book	0.8969±0.5576	0.1131±0.0398	0.8607±0.6089	0.1077±0.0532
Girl	0.0819±0.0092	0.0889±0.0083	0.0734±0.0067	0.0656±0.0084
Hand2	0.1210±0.0248	0.1162±0.0173	0.1026±0.0149	0.1095±0.0278
Helicopter	0.1112±0.0068	0.0986±0.0057	0.0906±0.0098	0.0883±0.0151
Polo	0.1242±0.0059	0.0998±0.0055	0.1132±0.0120	0.1085±0.0122
Road	0.2880±0.0684	0.2729±0.0673	0.2426±0.0791	0.2381±0.0808
Wheel	0.2278±0.0150	0.1941±0.0160	0.1872±0.0212	0.2258±0.0367

Table 3.3: Performance of camshift, WLT, WLTMS and WGGMMT in terms of size error (mean±std)

Seq.	camshift	WLT	WLTMS	WGGMMT
Book	0.1403±0.0789	0.0782±0.0304	0.1085±0.0490	0.1056±0.0482
Girl	0.1653±0.0196	0.1532±0.0159	0.1392±0.0137	0.1565±0.0115
Hand2	0.1872±0.0328	0.1397±0.0345	0.1330±0.0304	0.1713±0.0675
Helicopter	0.1085±0.0087	0.1185±0.0076	0.0998±0.0100	0.0990±0.0170
Polo	0.2241±0.0263	0.1764±0.0104	0.1817±0.0269	0.2786±0.0607
Road	0.2956±0.0660	0.2795±0.0528	0.2822±0.0603	0.3003±0.0599
Wheel	0.2603±0.0317	0.2380±0.0194	0.2315±0.0154	0.3249±0.0204

Table 3.4: Performance of camshift, WLT, WLTMS and WGGMMT in terms of average precision

Seq.	camshift	WLT	WLTMS	WGGMMT
Book	0.2338	0.9214	0.2416	0.9763
Girl	0.8649	0.8834	0.9152	0.9324
Hand2	0.8416	0.9428	0.9506	0.9810
Helicopter	0.8831	0.9135	0.9127	0.9507
Polo	0.8506	0.9316	0.9187	0.9290
Road	0.7692	0.8213	0.8205	0.8348
Wheel	0.7818	0.8492	0.8632	0.9653

Table 3.5: Performance of camshift, WLT, WLTMS and WGGMMT in terms of average recall

Seq.	camshift	WLT	WLTMS	WGGMMT
Book	0.1426	0.8821	0.1637	0.7231
Girl	0.8210	0.8497	0.8753	0.8547
Hand2	0.7321	0.7954	0.8127	0.6205
Helicopter	0.9108	0.9189	0.9326	0.8824
Polo	0.7652	0.8837	0.8643	0.5712
Road	0.7376	0.7732	0.7591	0.6886
Wheel	0.7544	0.8185	0.8202	0.3244

Table 3.6: Performance of camshift, WLT, WLTMS and WGGMMT in terms of average F-measure

Seq.	camshift	WLT	WLTMS	WGGMMT
Book	0.1780	0.8966	0.1992	0.8281
Girl	0.8416	0.8686	0.8918	0.8912
Hand2	0.7804	0.8584	0.8719	0.7636
Helicopter	0.8927	0.9145	0.9208	0.9157
Polo	0.8050	0.9058	0.8878	0.7099
Road	0.7518	0.7985	0.7936	0.7543
Wheel	0.7662	0.8324	0.8451	0.4901

lost their target midway, caused by the object's rapid drift and a region in the nearby background with close colour to the target (the likelihood falls into a local optimum and cannot jump out). The WLT and WGGMMT successfully track the target, but we notice that the size error of the WGGMM appears to rise and fall periodically. When the edges of the book are not parallel to the axes, the size update method makes the four corners of the book outside of the ellipse, which increases the size error. Although the girl in the second video sequence rotates in movement, the switch is very smooth, and the distribution of the main colour features of the target (yellow-green jacket and pink pants) almost does not change. Therefore, all four methods have sound effects. The WGGMM updates the model near the end of the sequence. At this moment, the primary colours of the other person adjacent to the target are black and white. But the aspect ratio of the ellipse representing the target has not changed dramatically. Accordingly, these new colours are still considered background. The main challenge given in video sequence three is the non-rigid deformation due to the dance actions of the characters. The WLTMS is better by one tally in this test. Our method underestimates the size of the target for the reasons specified earlier, particularly when the figure's legs open and the background fill. Similar situations appear in the series polo and wheel. Especially, WGGMMT does

a poor job of tracking cyclists, where the colour of the bike is very close to the background, and its proportion in the target area is small. Then the main subject being tracked becomes the rider, and the algorithm completely ignores the bike, resulting in a notable deviation in terms of position and size error. Other algorithms have the same problem, but not to the equivalent extent. In the helicopter sequence, the camera's perspective follows the target when shotting, so the background replaces significantly (from grass to blue sky). The WGGMMT has excellent performance (both in position and size) in this sequence because the shape of the central part of the helicopter is nearly egg-shaped. For the road sequence, its characteristic is that the target is tiny compared with the entire frame. As a result, the errors show a discrete and quantized variation. Due to the unique sampling method adopted by the size update mechanism of WLT and WLTMS, the minimum size of the ellipse is limited. In our comparison research, we use 11 symmetrical sampling lines in both the horizontal and vertical directions. It can be seen that there is a lot of platform-like waveform in error change in both of these two approaches. Fig. 3.16 displays the representative results of WGGMMT on the real datasets (every ten frames).

Figure 3.9: Performance (book) of camshift, WLT, WLTMS and WGGMMT in terms of position and size error

Figure 3.10: Performance (girl) of camshift, WLT, WLTMS and WGGMMT in terms of position and size error

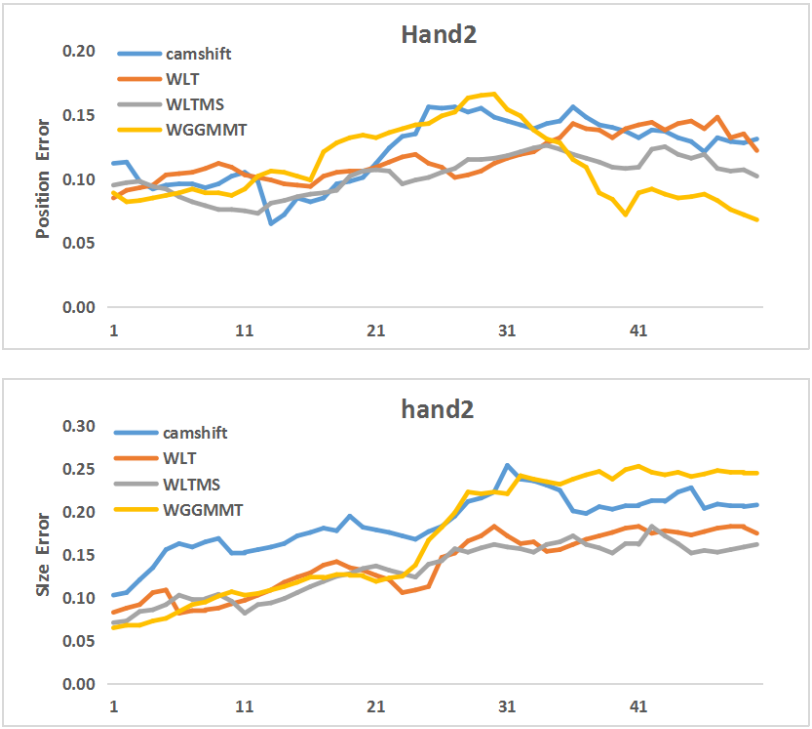


Figure 3.11: Performance (hand2) of camshift, WLT, WLTMS and WGGMMT in terms of position and size error

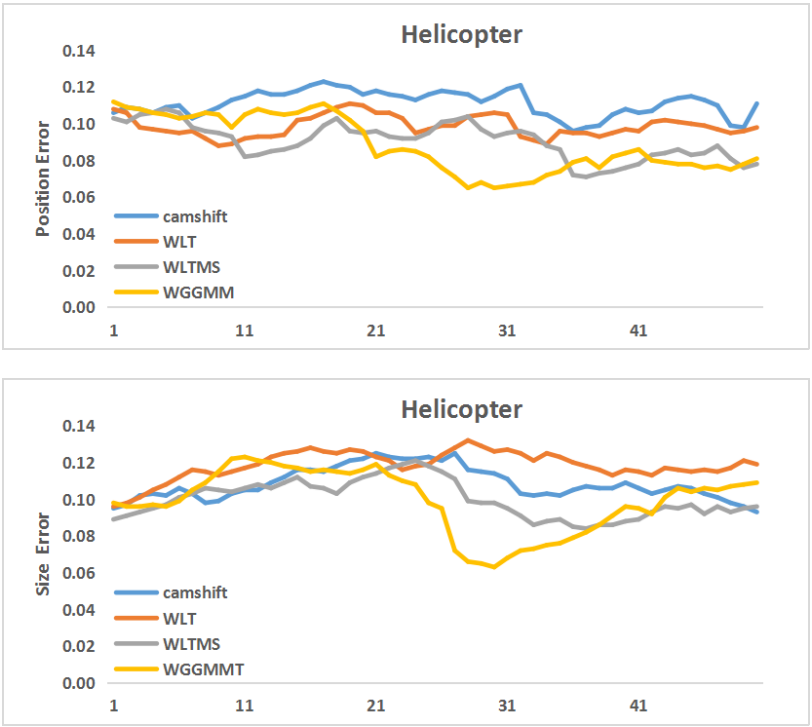


Figure 3.12: Performance (helicopter) of camshift, WLT, WLTMS and WGGMMT in terms of position and size error

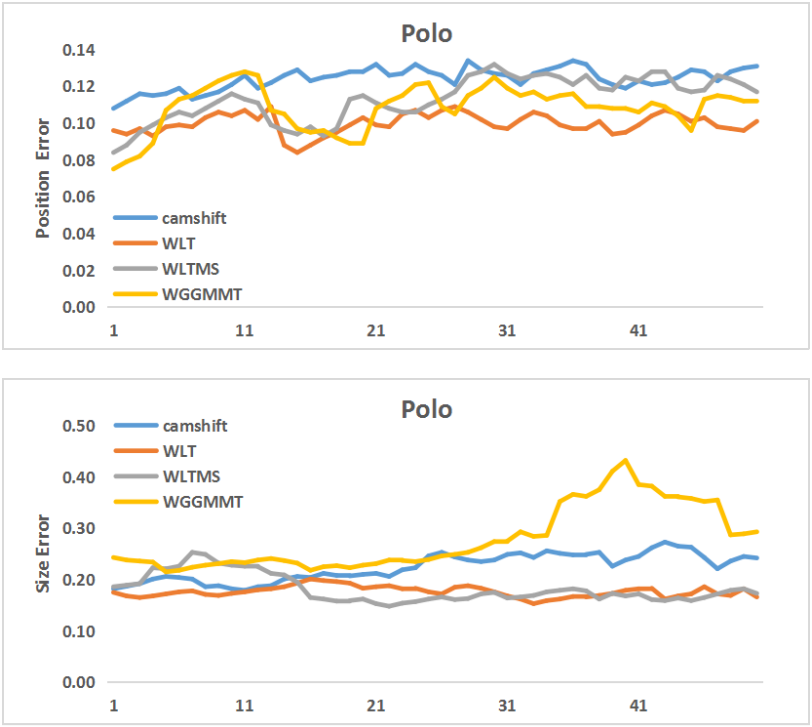


Figure 3.13: Performance (polo) of camshift, WLT, WLTMS and WGGMMT in terms of position and size error

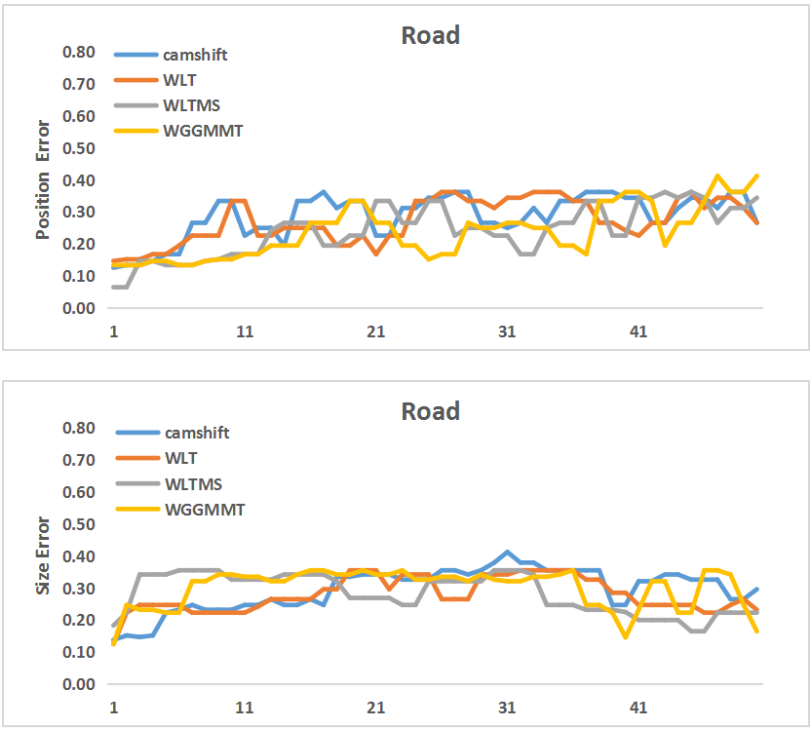


Figure 3.14: Performance (road) of camshift, WLT, WLTMS and WGGMMT in terms of position and size error

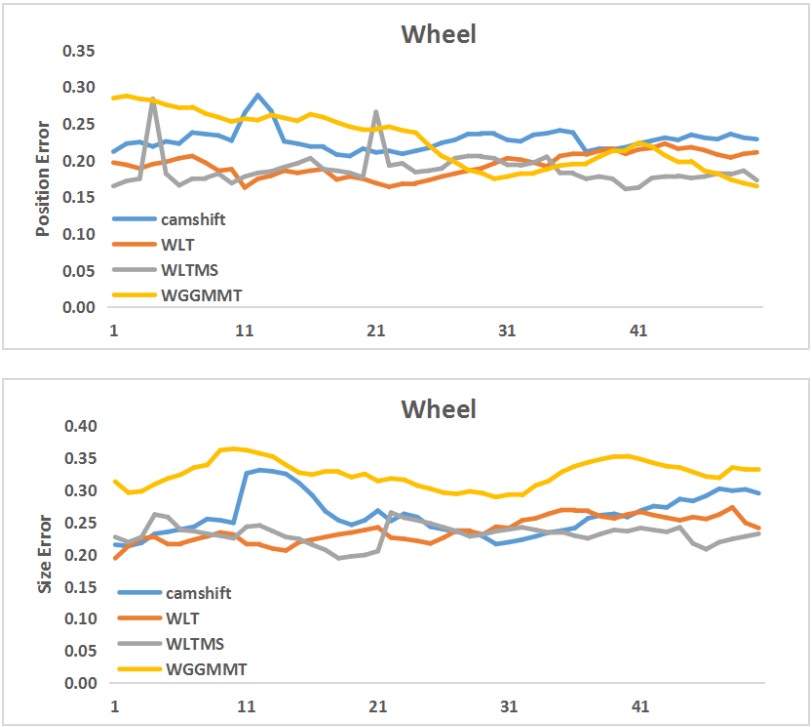


Figure 3.15: Performance (wheel) of camshift, WLT, WLTMS and WGGMMT in terms of position and size error

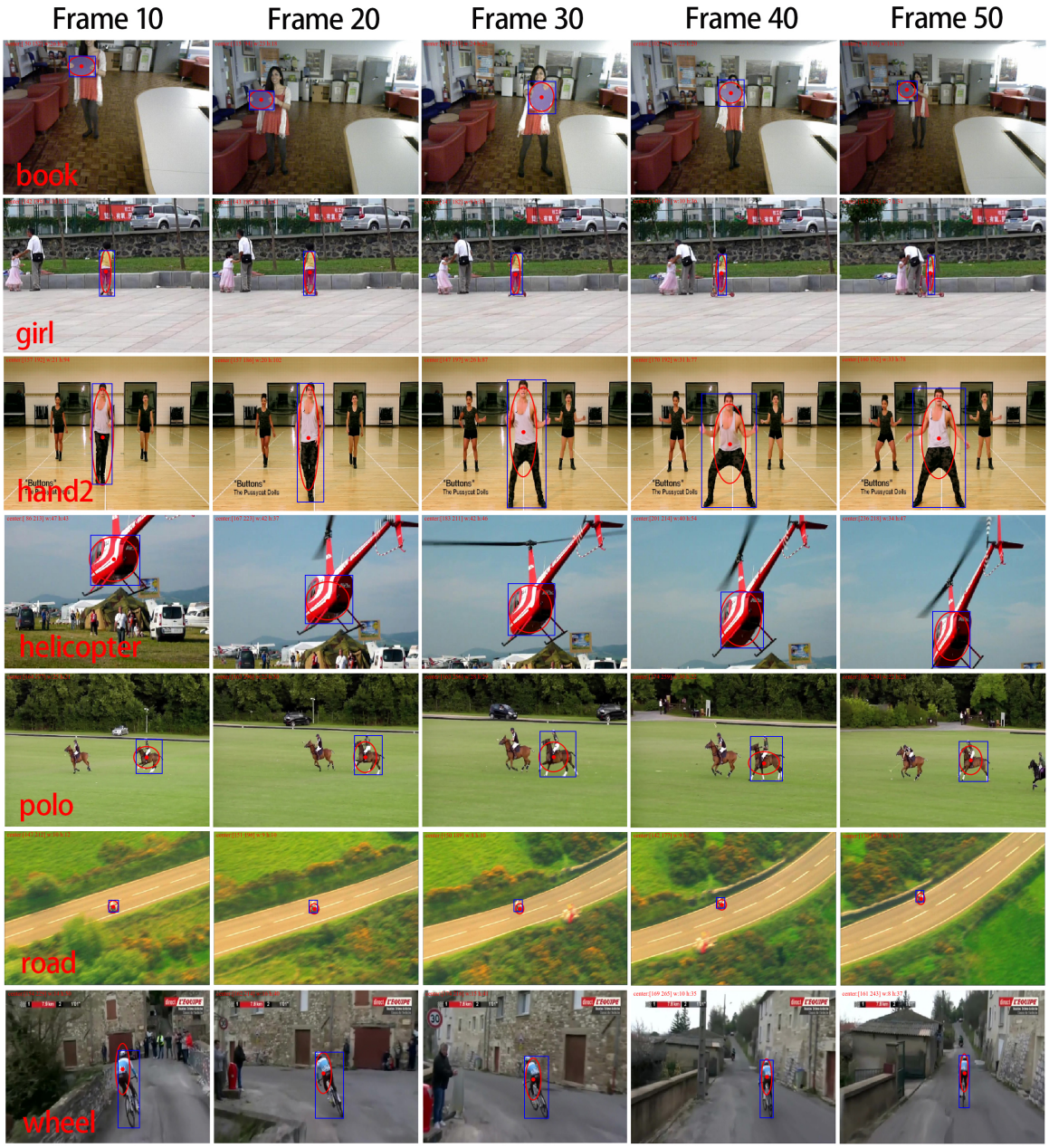


Figure 3.16: Representative results (red ellipse) using WGGMMT on the real datasets, the blue box is the ground truth

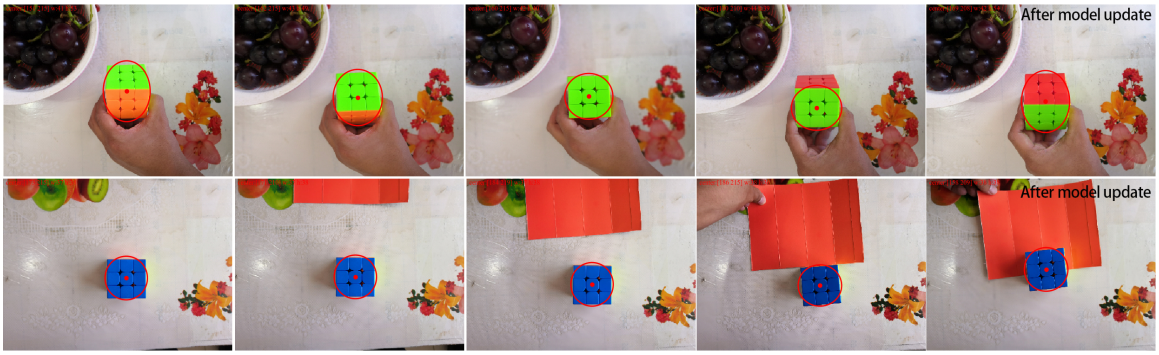


Figure 3.17: Representative results of model update when applying WGGMMT

Fig. 3.17 shows the qualitative results of WGGMMT in the model update. We use a Rubik's cube to create a typical scene. The tracking using the non-updated model is carried out during the cube rotation. The aspect ratio of the ellipse has a remarkable transformation in this process. Consequently, the components of new colours within the background ellipse are supplemented to the target model and the disappearing colour components are eliminated when modelling renewal. The newly added colours are also selected in the ellipse after the model updating. In contrast, the other condition is the added colour caused by the change of background around the target. Since the ellipse aspect ratio is almost identical, the target model remains the same after the update.

Without considering model initialization and update, the average time required by our algorithm to complete a frame tracking on a standard PC is about 45 ms, of which most time (about 33 ms on average) is spent in the clustering process of image colour compression (10 categories). However, when it comes to the model establishment since the parameters of the generalized Gaussian mixture model have no closed form, the numerical solution based on iteration will take 2-3 minutes. Therefore, matched with other techniques, this method still has some shortcomings in absolute real-time performance.

3.5 Conclusion

This paper proposes a tracking method based on the spatially weighted generalized Gaussian mixture model. In the preprocessing operation, colour compression enhances the GGMM's adaptability to target features while preserving the original colour distribution as far as possible. This

algorithm combines the idea of image segmentation and likelihood maximization to build the discriminative model and improve the mean-shift update method. The design of size estimation based on the sum of weight is used to solve the problem that the derivative of likelihood cannot be obtained. The model update is supported by the change of aspect ratio of the target ellipse, which further advances the algorithm's robustness. Experimental results show that the tracking accuracy of this method is better than that of other methods in most cases, and the size estimation is conservative. However, this method still has some weaknesses in real-time performance. Future work will attempt to reduce the time overhead of the algorithm and extend the framework to multi-target tracking.

Chapter 4

Conclusion and Future Work

4.1 Conclusion

This thesis proposes a weighted generalized Gaussian mixture model framework. The mean and covariance parameters of the mixture model components are obtained through fixed-point equations. The Newton-Raphson method is applied to get the shape parameters. The algorithm automatically determines the number of components according to the MML criterion. The weighting method dramatically enhances the contrast between the primary and secondary features of the data, thereby improving the model's adaptability to fit peak distributions. The framework can effectively remove the interference caused by outliers and noise and has satisfying robustness and accuracy. Due to the introduction of shape parameters, the method based on variational inference is not feasible because it is difficult to find the conjugate priors of the complete covariance. However, when using the MLE method to promote the EM algorithm, almost all model parameters do not have closed solutions. The iteration based on a numerical method causes a lot of time consumption, and the real-time performance of modelling is relatively poor. In addition, the calculation of shape parameters requires a certain degree of data support. Due to the truncated nature of computer floating-point operations, logarithmic overflow errors are prone to occur in the case of insufficient samples. The above framework is applied to robust point cloud registration and single target tracking. Experiments manifest that the proposed framework can adequately cope with the problems of different sampling rates and sensor noise in point cloud registration and accurately extract the structural features in the dense

area of the point cloud. After combining with stochastic optimization, it can significantly reduce the situation of falling into a local optimum. However, this method is only suitable for solid point cloud data, and it does not work well for contour or surface point clouds. In visual tracking, the framework joined with colour compression preprocessing can establish a more reliable discriminant model to distinguish between the target and the background. The logarithmic responsiveness ratio to the models provides segmentation weights for location and scale update, and the distance-based kernel weighting can adequately suppress the edge effect. The model updating strategy based on the ellipse aspect ratio change can effectively deal with the colour alteration caused by the three-dimensional rotation of the target. Compared with other algorithms, it is proved that the proposed method is superior in position accuracy but conservative in size estimation.

4.2 Future Work

Because of the shortcomings of existing work, we suggest the following prospective directions and possible improvements: (1) Aiming at the time performance problem of the generalized Gaussian mixture model. When using numerical methods to solve parameters, many formulas have the same sub-blocks. Therefore, we can directly store the first calculation results as variables through reasonable memory arrangements and call them in the subsequent process to reduce repeated computations. Furthermore, software or hardware acceleration can be applied, such as an optimized library of matrix operations to increase the speed of inversion and accumulation or using GPU for parallel computing. As far as the framework algorithm itself is concerned, it may be required to introduce other auxiliary methods (e.g. sampling) to roughly estimate the model's parameters and then use iterative approaches to make fine adjustments to speed up the convergence. For some specific applications, reducing the precision of iteration can improve efficiency. For example, when modelling the colour of an object, the colour space is discrete, and the step size is one. The accuracy of the model parameters is not necessarily valid to decimal places. (2) For the problems in the registration process of the point cloud. The proposed method models both the point cloud to be registered and the target point cloud and uses KLD to measure the difference between them to find the optimal transformation parameters. Since rigid registration only involves translation and rotation, the model

parameters of the transformed point cloud do not need to be recalculated. Instead, we can get them by applying the same transformation to the previous mixture model components. However, in the non-rigid registration, the mixture model must be re-established from the transformed point cloud, which means that parameter estimation will be carried out several times in the optimization process. This time consumption is unacceptable. Therefore, one possible approach is to model only the reference point cloud and use the maximum a posteriori idea to make the point cloud to be matched have the maximum likelihood in the target model, which is also applicable to non-rigid transformations. In addition, the proposed method is only suitable for solid point clouds. Since the three views of the object can describe the structural information, we can consider projecting the 3D point cloud to the plane with different perspectives and then using WMGGMM to represent the projection points to enhance the description ability of the surface point cloud. (3) For problems in visual tracking. In the proposed method, using k-means to compress each frame takes a lot of time. The improved idea is to perform colour compression on the ROI window instead of the entire frame. But the following problem is that as the pixels in the ROI change, the results of colour compressions may be inconsistent. The identical pixel in the original image may be assigned different values. Considering that the change between frames is not significant, we can perform colour compression at a specific interval. During this period, the principle of nearest neighbour classification is utilized to assign values to pixels based on the last colour compression results. This will reduce the amount of data involved in clustering and obtains the same contrast with a lower number of clusters.

Bibliography

[1] Frédéric Pascal, Lionel Bombrun, Jean-Yves Tourneret, and Yannick Berthoumieu. Parameter estimation for multivariate generalized gaussian distributions. *IEEE Transactions on Signal Processing*, 61(23):5960–5971, 2013.

[2] Mohand Said Allili, Nizar Bouguila, and Djemel Ziou. Finite general gaussian mixture modeling and application to image and video foreground segmentation. *Journal of Electronic Imaging*, 17(1):013005, 2008.

[3] Fatma Najar, Sami Bourouis, Nizar Bouguila, and Safya Belghith. Unsupervised learning of finite full covariance multivariate generalized gaussian mixture models for human activity recognition. *Multimedia Tools and Applications*, 78(13):18669–18691, 2019.

[4] Jingfan Fan, Jian Yang, Danni Ai, Likun Xia, Yitian Zhao, Xing Gao, and Yongtian Wang. Convex hull indexed gaussian mixture model (ch-gmm) for 3d point set registration. *Pattern Recognition*, 59:126–141, 2016.

[5] Vasileios Karavasilis, Christophoros Nikou, and Aristidis Likas. Visual tracking using spatially weighted likelihood of gaussian mixtures. *Computer vision and image understanding*, 140:43–57, 2015.

[6] M Th Subbotin. On the law of frequency of error. *Mathematicheskii Sbornik*, 31(2):296–301, 1923.

[7] Eusebio Gómez, MA Gomez-Viilegas, and J Miguel Marín. A multivariate generalization of the power exponential family of distributions. *Communications in Statistics-Theory and Methods*, 27(3):589–600, 1998.

[8] Eusebio Gómez Sánchez-Manzano, Miguel Angel Gomez-Villegas, and Juan-Miguel Marín-Diazaraque. A matrix variate generalization of the power exponential family of distributions. *Communications in Statistics-Theory and Methods*, 31(12):2167–2182, 2002.

[9] Kai-Sheng Song. A globally convergent and consistent method for estimating the shape parameter of a generalized gaussian distribution. *IEEE Transactions on Information Theory*, 52(2):510–527, 2006.

[10] Alex Dytso, Ronit Bustin, H Vincent Poor, and Shlomo Shamai. Analytical properties of generalized gaussian distributions. *Journal of Statistical Distributions and Applications*, 5(1):1–40, 2018.

[11] E Gómez-Sánchez-Manzano, MA Gómez-Villegas, and JM Marín. Multivariate exponential power distributions as mixtures of normal distributions with bayesian applications. *Communications in Statistics—Theory and Methods*, 37(6):972–985, 2008.

[12] Esa Ollila, David E Tyler, Visa Koivunen, and H Vincent Poor. Complex elliptically symmetric distributions: Survey, new results and applications. *IEEE Transactions on signal processing*, 60(11):5597–5625, 2012.

[13] Mahesh K Varanasi and Behnaam Aazhang. Parametric generalized gaussian density estimation. *The Journal of the Acoustical Society of America*, 86(4):1404–1415, 1989.

[14] Ricardo Antonio Maronna. Robust m-estimators of multivariate location and scatter. *The annals of statistics*, pages 51–67, 1976.

[15] Frédéric Pascal, Yacine Chitour, Jean-Philippe Ovarlez, Philippe Forster, and Pascal Larzabal. Covariance structure maximum-likelihood estimates in compound gaussian noise: Existence and algorithm analysis. *IEEE Transactions on Signal Processing*, 56(1):34–48, 2007.

[16] Yacine Chitour and Frédéric Pascal. Exact maximum likelihood estimates for sirv covariance matrix: Existence and algorithm analysis. *IEEE Transactions on signal processing*, 56(10):4563–4573, 2008.

[17] Teng Zhang, Ami Wiesel, and Maria Sabrina Greco. Multivariate generalized gaussian distribution: Convexity and graphical models. *IEEE Transactions on Signal Processing*, 61(16):4141–4148, 2013.

[18] Graciela Gonzalez-Farias, J Armando Dominguez Molina, and Ramón M Rodríguez-Dagnino. Efficiency of the approximated shape parameter estimator in the generalized gaussian distribution. *IEEE Transactions on Vehicular Technology*, 58(8):4214–4223, 2009.

[19] Muhammad Azam and Nizar Bouguila. Bounded generalized gaussian mixture model with ICA. *Neural Process. Lett.*, 49(3):1299–1320, 2019.

[20] Muhammad Azam and Nizar Bouguila. Unsupervised keyword spotting using bounded generalized gaussian mixture model with ICA. In *2015 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2015, Orlando, FL, USA, December 14-16, 2015*, pages 1150–1154. IEEE, 2015.

[21] Tarek Elguebaly and Nizar Bouguila. Semantic scene classification with generalized gaussian mixture models. In Mohamed Kamel and Aurélio J. C. Campilho, editors, *Image Analysis and Recognition - 12th International Conference, ICIAR 2015, Niagara Falls, ON, Canada, July 22-24, 2015, Proceedings*, volume 9164 of *Lecture Notes in Computer Science*, pages 159–166. Springer, 2015.

[22] Tarek Elguebaly and Nizar Bouguila. Bayesian learning of generalized gaussian mixture models on biomedical images. In Friedhelm Schwenker and Neamat El Gayar, editors, *Artificial Neural Networks in Pattern Recognition, 4th IAPR TC3 Workshop, ANNPR 2010, Cairo, Egypt, April 11-13, 2010. Proceedings*, volume 5998 of *Lecture Notes in Computer Science*, pages 207–218. Springer, 2010.

[23] Tarek Elguebaly and Nizar Bouguila. Background subtraction using finite mixtures of asymmetric gaussian distributions and shadow detection. *Mach. Vis. Appl.*, 25(5):1145–1162, 2014.

[24] Tarek Elguebaly and Nizar Bouguila. Generalized gaussian mixture models as a nonparametric bayesian approach for clustering using class-specific visual features. *J. Vis. Commun. Image Represent.*, 23(8):1199–1212, 2012.

[25] Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. An introduction to mcmc for machine learning. *Machine learning*, 50(1):5–43, 2003.

[26] Srikanth Amudala, Samr Ali, Fatma Najar, and Nizar Bouguila. Variational inference of finite generalized gaussian mixture models. In *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 2433–2439. IEEE, 2019.

[27] Srikanth Amudala, Samr Ali, and Nizar Bouguila. Variational inference of infinite generalized gaussian mixture models with feature selection. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 120–127. IEEE, 2020.

[28] Fatma Najar, Sami Bourouis, Nizar Bouguila, and Safya Belghith. Unsupervised learning of finite full covariance multivariate generalized gaussian mixture models for human activity recognition. *Multimedia Tools and Applications*, 78(13):18669–18691, 2019.

[29] Shu-Kai S Fan and Yen Lin. A fast estimation method for the generalized gaussian mixture distribution on complex images. *Computer Vision and Image Understanding*, 113(7):839–853, 2009.

[30] Xuejing Gu, Xu Wang, and Yucheng Guo. A review of research on point cloud registration methods. In *IOP Conference Series: Materials Science and Engineering*, volume 782, pages 022–070. IOP Publishing, 2020.

[31] Dror Aiger, Niloy J Mitra, and Daniel Cohen-Or. 4-points congruent sets for robust pairwise surface registration. In *ACM SIGGRAPH 2008 papers*, pages 1–10. 2008.

[32] Nicolas Mellado, Dror Aiger, and Niloy J Mitra. Super 4pcs fast global pointcloud registration via smart indexing. In *Computer Graphics Forum*, volume 33, pages 205–215. Wiley Online Library, 2014.

[33] Pascal W Theiler, Jan D Wegner, and Konrad Schindler. Markerless point cloud registration with keypoint-based 4-points congruent sets. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 1(2):283–288, 2013.

[34] Xuming Ge. Automatic markerless registration of point clouds with semantic-keypoint-based 4-points congruent sets. *ISPRS Journal of Photogrammetry and Remote Sensing*, 130:344–357, 2017.

[35] Mustafa Mohamad, David Rappaport, and Michael Greenspan. Generalized 4-points congruent sets for 3d registration. In *2014 2nd international conference on 3D vision*, volume 1, pages 83–90. IEEE, 2014.

[36] Peter Biber and Wolfgang Straßer. The normal distributions transform: A new approach to laser scan matching. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, volume 3, pages 2743–2748. IEEE, 2003.

[37] Martin Magnusson, Achim Lilienthal, and Tom Duckett. Scan registration for autonomous mining vehicles using 3d-ndt. *Journal of Field Robotics*, 24(10):803–827, 2007.

[38] Cihan Ulaş and Hakan Temeltaş. A fast and robust scan matching algorithm based on ml-ndt and feature extraction. In *2011 IEEE International Conference on Mechatronics and Automation*, pages 1751–1756. IEEE, 2011.

[39] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE international conference on robotics and automation*, pages 3212–3217. IEEE, 2009.

[40] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992.

[41] Soon-Yong Park and Murali Subbarao. An accurate and fast point-to-plane registration technique. *Pattern Recognition Letters*, 24(16):2967–2976, 2003.

[42] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-icp. In *Robotics: science and systems*, volume 2, page 435. Seattle, WA, 2009.

[43] Jacopo Serafin and Giorgio Grisetti. Nicp: Dense normal based point cloud registration. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 742–749. IEEE, 2015.

[44] Andriy Myronenko and Xubo Song. Point set registration: Coherent point drift. *IEEE transactions on pattern analysis and machine intelligence*, 32(12):2262–2275, 2010.

[45] Peng Wang, Ping Wang, ZhiGuo Qu, YingHui Gao, and ZhenKang Shen. A refined coherent point drift (cpd) algorithm for point set registration. *Science China Information Sciences*, 54(12):2639–2646, 2011.

[46] Osamu Hirose. A bayesian formulation of coherent point drift. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[47] Hanxuan Yang, Ling Shao, Feng Zheng, Liang Wang, and Zhan Song. Recent advances and trends in visual tracking: A review. *Neurocomputing*, 74(18):3823–3831, 2011.

[48] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002.

[49] Jifeng Ning, Lei Zhang, David Zhang, and Chengke Wu. Scale and orientation adaptive mean shift tracking. *IET Computer Vision*, 6(1):52–61, 2012.

[50] Tomas Vojir, Jana Noskova, and Jiri Matas. Robust scale-adaptive mean-shift for tracking. *Pattern Recognition Letters*, 49:250–258, 2014.

[51] YT Chan, AGC Hu, and JB Plant. A kalman filter based tracking scheme with input estimation. *IEEE transactions on Aerospace and Electronic Systems*, (2):237–244, 1979.

[52] Kenji Okuma, Ali Taleghani, Nando De Freitas, James J Little, and David G Lowe. A boosted particle filter: Multitarget detection and tracking. In *European conference on computer vision*, pages 28–39. Springer, 2004.

[53] David S Bolme, J Ross Beveridge, Bruce A Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 2544–2550. IEEE, 2010.

[54] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE transactions on pattern analysis and machine intelligence*, 37(3):583–596, 2014.

[55] Martin Danelljan, Gustav Häger, Fahad Khan, and Michael Felsberg. Accurate scale estimation for robust visual tracking. In *British Machine Vision Conference, Nottingham, September 1-5, 2014*. BMVA Press, 2014.

[56] Yang Li and Jianke Zhu. A scale adaptive kernel correlation filter tracker with feature integration. In *European conference on computer vision*, pages 254–265. Springer, 2014.

[57] Peixia Li, Dong Wang, Lijun Wang, and Huchuan Lu. Deep visual tracking: Review and experimental comparison. *Pattern Recognition*, 76:323–338, 2018.

[58] Martin Danelljan, Andreas Robinson, Fahad Shahbaz Khan, and Michael Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In *European conference on computer vision*, pages 472–488. Springer, 2016.

[59] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Eco: Efficient convolution operators for tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6638–6646, 2017.

[60] Naiyan Wang and Dit Yan Yeung. Learning a deep compact image representation for visual tracking. *Advances in neural information processing systems*, 2013.

[61] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*, pages 850–865. Springer, 2016.

[62] Jack Valmadre, Luca Bertinetto, Joao Henriques, Andrea Vedaldi, and Philip HS Torr. End-to-end representation learning for correlation filter based tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2805–2813, 2017.

[63] Hao Zhu, Bin Guo, Ke Zou, Yongfu Li, Ka-Veng Yuen, Lyudmila Mihaylova, and Henry Leung. A review of point set registration: From pairwise registration to groupwise registration. *Sensors*, 19(5):1191, 2019.

[64] V De Silva, J Roche, and A Kondoz. Fusion of lidar and camera sensor data for environment sensing in driverless vehicles. arxiv 2018. *arXiv preprint arXiv:1710.06230*, 2018.

[65] Michael Giering, Vivek Venugopalan, and Kishore Reddy. Multi-modal sensor registration for vehicle perception via deep neural networks. In *2015 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–6. IEEE, 2015.

[66] Andrew Mastin, Jeremy Kepner, and John Fisher. Automatic registration of lidar and optical images of urban scenes. In *2009 IEEE conference on computer vision and pattern recognition*, pages 2639–2646. IEEE, 2009.

[67] Vinicio Rosas-Cervantes and Soon-Geul Lee. 3d localization of a mobile robot by using monte carlo algorithm and 2d features of 3d point cloud. *International Journal of Control, Automation and Systems*, pages 1–11, 2020.

[68] Weixin Lu, Guowei Wan, Yao Zhou, Xiangyu Fu, Pengfei Yuan, and Shiyu Song. Deepvcp: An end-to-end deep neural network for point cloud registration. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 12–21, 2019.

[69] Leonardo Rundo, Andrea Tangherloni, Carmelo Militello, Maria Carla Gilardi, and Giancarlo Mauri. Multimodal medical image registration using particle swarm optimization: A review. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8. IEEE, 2016.

[70] André Collignon, Dirk Vandermeulen, Paul Suetens, and Guy Marchal. 3d multi-modality medical image registration using feature space clustering. In *International Conference on Computer Vision, Virtual Reality, and Robotics in Medicine*, pages 195–204. Springer, 1995.

[71] Martin Sinko, Patrik Kamencay, Robert Hudec, and Miroslav Benco. 3d registration of the point cloud data using icp algorithm in medical image analysis. In *2018 ELEKTRO*, pages 1–6. IEEE, 2018.

[72] Sabry F El-Hakim, J-A Beraldin, Michel Picard, and Guy Godin. Detailed 3d reconstruction of large-scale heritage sites with integrated techniques. *IEEE Computer Graphics and Applications*, 24(3):21–29, 2004.

[73] Jiayi Ma, Junjun Jiang, Chengyin Liu, and Yansheng Li. Feature guided gaussian mixture model with semi-supervised em and local geometric constraint for retinal image registration. *Information Sciences*, 417:128–142, 2017.

[74] Maik Keller, Damien Lefloch, Martin Lambers, Shahram Izadi, Tim Weyrich, and Andreas Kolb. Real-time 3d reconstruction in dynamic scenes using point-based fusion. In *2013 International Conference on 3D Vision-3DV 2013*, pages 1–8. IEEE, 2013.

[75] Zhengyou Zhang. Iterative point matching for registration of free-form curves and surfaces. *International journal of computer vision*, 13(2):119–152, 1994.

[76] Donatello Conte, Pasquale Foggia, Carlo Sansone, and Mario Vento. Thirty years of graph matching in pattern recognition. *International journal of pattern recognition and artificial intelligence*, 18(03):265–298, 2004.

[77] Wei Gao and Russ Tedrake. Filterreg: Robust and efficient probabilistic point-set registration using gaussian filter and twist parameterization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11095–11104, 2019.

[78] Bing Jian and Baba C Vemuri. Robust point set registration using gaussian mixture models. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1633–1645, 2010.

[79] Wenbing Tao and Kun Sun. Asymmetrical gauss mixture models for point sets matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1598–1605, 2014.

[80] Nishant Ravikumar, Ali Gooya, Serkan Çimen, Alejandro F Frangi, and Zeike A Taylor. Group-wise similarity registration of point sets using student's t-mixture model for statistical shape models. *Medical image analysis*, 44:156–176, 2018.

[81] Simon Franchini, Alexandros Charogiannis, Christos N Markides, Martin J Blunt, and Samuel Krevor. Calibration of astigmatic particle tracking velocimetry based on generalized gaussian feature extraction. *Advances in Water Resources*, 124:1–8, 2019.

[82] Yuki Kubo, Norihiro Takamune, Daichi Kitamura, and Hiroshi Saruwatari. Blind speech extraction based on rank-constrained spatial covariance matrix estimation with multivariate generalized gaussian distribution. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:1948–1963, 2020.

[83] Hristina Hristova, Olivier Le Meur, Remi Cozot, and Kadi Bouatouch. Transformation of the multivariate generalized gaussian distribution for image editing. *IEEE transactions on visualization and computer graphics*, 24(10):2813–2826, 2017.

[84] Geert Verdoolaege and Paul Scheunders. Geodesics on the manifold of multivariate generalized gaussian distributions with an application to multicomponent texture discrimination. *International Journal of Computer Vision*, 95(3):265, 2011.

[85] Hassan Rami, Leila Belmerhnia, Ahmed Drissi El Maliani, and Mohammed El Hassouni. Texture retrieval using mixtures of generalized gaussian distribution and cauchy–schwarz divergence in wavelet domain. *Signal Processing: Image Communication*, 42:45–58, 2016.

[86] Geert Verdoolaege, Yves Rosseel, Michiel Lambrechts, and Paul Scheunders. Wavelet-based colour texture retrieval using the kullback-leibler divergence between bivariate generalized gaussian models. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 265–268. IEEE, 2009.

[87] Ines Channoufi, Sami Bourouis, Nizar Bouguila, and Kamel Hamrouni. Image and video denoising by combining unsupervised bounded generalized gaussian mixture modeling and spatial information. *Multimedia Tools and Applications*, 77(19):25591–25606, 2018.

[88] Ines Channoufi, Sami Bourouis, Nizar Bouguila, and Kamel Hamrouni. Color image segmentation with bounded generalized gaussian mixture model and feature selection. In *2018 4th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, pages 1–6. IEEE, 2018.

[89] Fatma Najar, Sami Bourouis, Nizar Bouguila, and Safya Belghith. A fixed-point estimation algorithm for learning the multivariate ggmm: application to human action recognition. In *2018 IEEE Canadian Conference on Electrical & Computer Engineering (CCECE)*, pages 1–4. IEEE, 2018.

[90] Israel Dejene Gebru, Xavier Alameda-Pineda, Florence Forbes, and Radu Horaud. Em algorithms for weighted-data clustering with application to audio-visual scene analysis. *IEEE transactions on pattern analysis and machine intelligence*, 38(12):2402–2415, 2016.

[91] Praveen Agarwal, Mohamed Jleli, and Bessem Samet. Banach contraction principle and applications. In *Fixed Point Theory in Metric Spaces*, pages 1–23. Springer, 2018.

[92] Mario A. T. Figueiredo and Anil K. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on pattern analysis and machine intelligence*, 24(3):381–396, 2002.

[93] John R Hershey and Peder A Olsen. Approximating the kullback leibler divergence between gaussian mixture models. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pages IV–317. IEEE, 2007.

[94] J-L Durrieu, J-Ph Thiran, and Finnian Kelly. Lower and upper bounds for approximation of the kullback-leibler divergence between gaussian mixture models. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4833–4836. Ieee, 2012.

[95] Shiyong Cui and Mihai Datcu. Comparison of kullback-leibler divergence approximation methods between gaussian mixture models for satellite image retrieval. In *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 3719–3722. IEEE, 2015.

[96] Mustansar Fiaz, Arif Mahmood, Sajid Javed, and Soon Ki Jung. Handcrafted and deep trackers: Recent visual object tracking approaches and trends. *ACM Computing Surveys (CSUR)*, 52(2):1–44, 2019.

[97] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis And Machine Intelligence*, 37(9), 2015.

[98] Gaocheng Liu, Shuai Liu, Khan Muhammad, Arun Kumar Sangaiah, and Faiyaz Doctor. Object tracking in vary lighting conditions for fog based intelligent surveillance of public spaces. *IEEE Access*, 6:29283–29296, 2018.

[99] Zheng Pan, Shuai Liu, Arun Kumar Sangaiah, and Khan Muhammad. Visual attention feature (vaf): a novel strategy for visual tracking based on cloud platform in intelligent surveillance systems. *Journal of Parallel and Distributed Computing*, 120:182–194, 2018.

[100] Alexander Buyval, Aidar Gabdullin, Ruslan Mustafin, and Ilya Shimchik. Realtime vehicle and pedestrian tracking for didi udacity self-driving car challenge. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2064–2069. IEEE, 2018.

[101] Kuan-Hui Lee, Jenq-Neng Hwang, Greg Okopal, and James Pitton. Ground-moving-platform-based human tracking using visual slam and constrained multiple kernels. *IEEE transactions on intelligent transportation systems*, 17(12):3602–3612, 2016.

[102] Redouane Khemmar, Matthias Gouveia, Benoît Decoux, and Jean-Yves Ertaud. Real time pedestrian and object detection and tracking-based deep learning. application to drone visual tracking. 2019.

[103] Xiaoyue Zhao, Fangling Pu, Zhihang Wang, Hongyu Chen, and Zhaozhuo Xu. Detection, tracking, and geolocation of moving vehicle from uav using monocular camera. *IEEE Access*, 7:101160–101170, 2019.

[104] Seyed Mojtaba Marvasti-Zadeh, Li Cheng, Hossein Ghanei-Yakhdan, and Shohreh Kasaei. Deep learning for visual tracking: A comprehensive survey. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[105] Yasuharu Kunii, Gabor Kovacs, and Naoaki Hoshi. Mobile robot navigation in natural environments using robust object tracking. In *2017 IEEE 26th international symposium on industrial electronics (ISIE)*, pages 1747–1752. IEEE, 2017.

[106] Mauro Fernandez-Sanjurjo, Brais Bosquet, Manuel Mucientes, and Victor M Brea. Real-time visual detection and tracking system for traffic monitoring. *Engineering Applications of Artificial Intelligence*, 85:410–420, 2019.

[107] Fozia Mehboob, Muhammad Abbas, Richard Jiang, Abdul Rauf, Shoab A Khan, and Saad Rehman. Trajectory based vehicle counting and anomalous event visualization in smart cities. *Cluster Computing*, 21(1):443–452, 2018.

[108] Naiyan Wang, Jianping Shi, Dit-Yan Yeung, and Jiaya Jia. Understanding and diagnosing visual tracking systems. In *Proceedings of the IEEE international conference on computer vision*, pages 3101–3109, 2015.

[109] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4):13–es, 2006.

[110] David A Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang. Incremental learning for robust visual tracking. *International journal of computer vision*, 77(1-3):125–141, 2008.

[111] Xu Jia, Huchuan Lu, and Ming-Hsuan Yang. Visual tracking via adaptive structural local sparse appearance model. In *2012 IEEE Conference on computer vision and pattern recognition*, pages 1822–1829. IEEE, 2012.

[112] Patrick Pérez, Carine Hue, Jaco Vermaak, and Michel Gangnet. Color-based probabilistic tracking. In *European Conference on Computer Vision*, pages 661–675. Springer, 2002.

[113] Amit Adam, Ehud Rivlin, and Ilan Shimshoni. Robust fragments-based tracking using the integral histogram. In *2006 IEEE Computer society conference on computer vision and pattern recognition (CVPR'06)*, volume 1, pages 798–805. IEEE, 2006.

[114] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Real-time tracking of non-rigid objects using mean shift. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 2, pages 142–149. IEEE, 2000.

[115] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Kernel-based object tracking. *IEEE Transactions on pattern analysis and machine intelligence*, 25(5):564–577, 2003.

[116] Katja Nummiaro, Esther Koller-Meier, and Luc Van Gool. An adaptive color-based particle filter. *Image and vision computing*, 21(1):99–110, 2003.

[117] Waqas Hassan, Nagachetan Bangalore, Philip Birch, Rupert Young, and Chris Chatwin. An adaptive sample count particle filter. *Computer Vision and Image Understanding*, 116(12):1208–1222, 2012.

[118] Joao F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *European conference on computer vision*, pages 702–715. Springer, 2012.

[119] Yibing Song, Chao Ma, Lijun Gong, Jiawei Zhang, Rynson WH Lau, and Ming-Hsuan Yang. Crest: Convolutional residual learning for visual tracking. In *Proceedings of the IEEE international conference on computer vision*, pages 2555–2564, 2017.

[120] Hyeonseob Nam, Mooyeol Baek, and Bohyung Han. Modeling and propagating cnns in a tree structure for visual tracking. *arXiv preprint arXiv:1608.07242*, 2016.

[121] Lijun Wang, Wanli Ouyang, Xiaogang Wang, and Huchuan Lu. Visual tracking with fully convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 3119–3127, 2015.

[122] Zhen Cui, Shengtao Xiao, Jiashi Feng, and Shuicheng Yan. Recurrently target-attending tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1449–1458, 2016.

[123] Martin Danelljan, Luc Van Gool, and Radu Timofte. Probabilistic regression for visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7183–7192, 2020.

[124] Zedu Chen, Bineng Zhong, Guorong Li, Shengping Zhang, and Rongrong Ji. Siamese box adaptive network for visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6668–6677, 2020.

[125] Paul Voigtlaender, Jonathon Luiten, Philip HS Torr, and Bastian Leibe. Siam r-cnn: Visual tracking by re-detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6578–6588, 2020.

[126] C-H Chou and K-C Liu. Colour image compression based on the measure of just noticeable colour difference. *IET Image Processing*, 2(6):304–322, 2008.

[127] A Messaoudi and K Srairi. Colour image compression algorithm based on the dct transform using difference lookup table. *Electronics Letters*, 52(20):1685–1686, 2016.

[128] Majid Rabbani. Jpeg2000: Image compression fundamentals, standards and practice. *Journal of Electronic Imaging*, 11(2):286, 2002.

[129] Ayan Banerjee and Amiya Halder. An efficient image compression algorithm for almost dual-color image based on k-means clustering, bit-map generation and rle. In *2010 International Conference on Computer and Communication Technology (ICCCT)*, pages 201–205. IEEE, 2010.

[130] G Vimala Kumari, G Sasibhushana Rao, and B Prabhakara Rao. Flower pollination-based k-means algorithm for medical image compression. *International Journal of Advanced Intelligence Paradigms*, 18(2):171–192, 2021.

[131] Douglas Kelker. Distribution theory of spherical distributions and a location-scale parameter generalization. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 419–430, 1970.

[132] Kai-Tai Fang, Samuel Kotz, and Kai Wang Ng. *Symmetric multivariate and related distributions*. Chapman and Hall/CRC, 2018.

[133] Nafaa Nacereddine, Aicha Baya Goumeidane, and Djemel Ziou. Unsupervised weld defect classification in radiographic images using multivariate generalized gaussian mixture model with exact computation of mean and shape parameters. *Computers in Industry*, 108:132–149, 2019.

[134] Fatma Najar, Sami Bourouis, Nizar Bouguila, and Safya Belghith. A new hybrid discriminative/generative model using the full-covariance multivariate generalized gaussian mixture models. *Soft Computing*, 24(14):10611–10628, 2020.

[135] Fatma Najar, Sami Bourouis, Atef Zaguia, Nizar Bouguila, and Safya Belghith. Unsupervised human action categorization using a riemannian averaged fixed-point learning of multivariate ggmm. In *International Conference Image Analysis and Recognition*, pages 408–415. Springer, 2018.

[136] V Sailaja, K Srinivasa Rao, and KVVS Reddy. Text independent speaker identification with finite multivariate generalized gaussian mixture model and hierarchical clustering algorithm. *Int. Journal of Computer Applications*, 11(11):0975–8887, 2010.

[137] Zois Boukouvalas, Geng-Shen Fu, and Tülay Adalı. An efficient multivariate generalized gaussian distribution estimator: Application to iva. In *2015 49th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–4. IEEE, 2015.

[138] Charles-Alban Deledalle, Shibin Parameswaran, and Truong Q Nguyen. Image denoising with generalized gaussian mixture model patch priors. *SIAM Journal on Imaging Sciences*, 11(4):2568–2609, 2018.

[139] K Naveen Kumar, K Srinivasa Rao, Y Srinivas, and Ch Satyanarayana. Texture segmentation based on multivariate generalized gaussian mixture model. *CMES: Computer Modeling in Engineering & Sciences*, 107(3):201–221, 2015.

[140] Guanglei Xiong, Chao Feng, and Liang Ji. Dynamical gaussian mixture model for tracking elliptical living objects. *Pattern Recognition Letters*, 27(7):838–842, 2006.

[141] RK Meghana, Yojan Chitkara, S Apoorva, et al. Background-modelling techniques for foreground detection and tracking using gaussian mixture model. In *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, pages 1129–1134. IEEE, 2019.

[142] LONG Hao and ZHANG Shu-kui. Moving object tracking algorithm based on improved gaussian mixture model. In *2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE)*, pages 271–275. IEEE, 2019.

[143] Harihara Santosh Dadi, Gopala Krishna Mohan Pillutla, and Madhavi Latha Makkena. Face recognition and human tracking using gmm, hog and svm in surveillance videos. *Annals of Data Science*, 5(2):157–179, 2018.

[144] Qi Zhao, Zhi Yang, and Hai Tao. Differential earth mover's distance with its applications to visual tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(2):274–287, 2008.