Motion segmentation of RGB-D sequences: Combining semantic and motion information using statistical inference

Sundaram Muthu[1], Ruwan Tennakoon[2], Tharindu Rathnayake[1], Reza Hoseinnezhad[1], David Suter[3], Alireza Bab-Hadiashar[1]

***Abstract*—This paper presents an innovative method for motion segmentation in RGB-D dynamic videos with multiple moving objects. The focus is on finding static, small or slow moving objects (often overlooked by other methods) that their inclusion can improve the motion segmentation results. In our approach, semantic object based segmentation and motion cues are combined to estimate the number of moving objects, their motion parameters and perform segmentation. Selective object-based sampling and correspondence matching are used to estimate object specific motion parameters. The main issue with such an approach is the over segmentation of moving parts due to the fact that different objects can have the same motion (e.g. background objects). To resolve this issue, we propose to identify objects with similar motions by characterizing each motion by a distribution of a simple metric and using a statistical inference theory to assess their similarities. To demonstrate the significance of the proposed statistical inference, we present an ablation study, with and without static objects inclusion, on SLAM accuracy using the TUM-RGBD dataset. To test the effectiveness of the proposed method for finding small or slow moving objects, we applied the method to RGB-D MultiBody and SBM-RGBD motion segmentation datasets. The results showed that we can improve the accuracy of motion segmentation for small objects while remaining competitive on overall measures.**

***Index Terms*—RGB-D Motion Segmentation, Multibody Structure and Motion, Dynamic SLAM, EVT, Kolmogorov–Smirnov test**

I. Introduction

Segmentation of independently moving objects in a complex dynamic scene, and estimation of their individual motion parameters (i.e. motion segmentation) are important tasks in many computer vision applications. Applications of motion segmentation include autonomous navigation [1], path planning [2], obstacle avoidance, surveillance and tracking in robotics or in autonomous driving [3]. Motion segmentation is also currently used in applications such as augmented reality [4] and scene flow estimation [5].

The complexity in motion segmentation is analogous to the chicken-and-egg problem. If the model parameters for each motion was known a priori, it would be trivial to derive the correct segmentation of data. If the segmentation is known, the model parameter estimation can be easily performed. However, solving the motion segmentation problem is challenging when both the motion models, and the segmentation, is unknown. The fact that available data is often inaccurate, and has high percentages of outliers (leading to wrong data association), adds to the challenge [6].

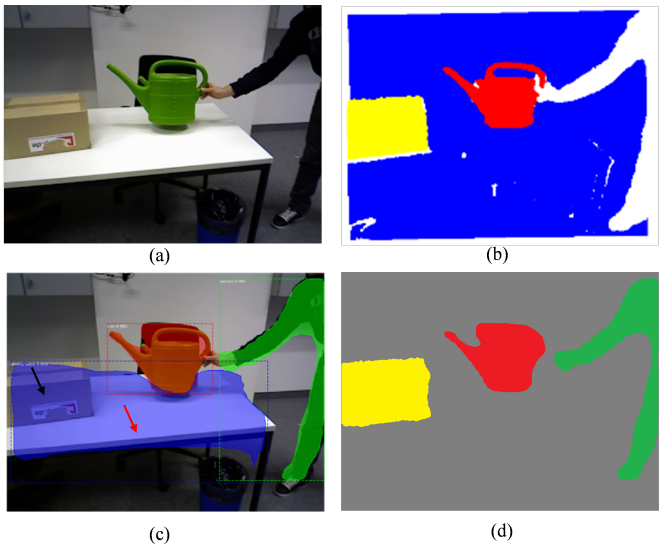


Fig. 1: A simple motion segmentation example to show the challenges of using object segmentation as a prior for motion segmentation. (a) Sample image (b) Ground truth segmentation (c) Mask-RCNN object segmentation and (d) the output of our method. The *black arrow* in (*c*) shows a missed detection (*the box* is not detected even though it is moving because it is not one of the training object classes) and the *red arrow* shows over-segmentation (*the table* that belong to background motion is assigned a separate label)

Segmentation of small or slow moving objects in a complex dynamic scene remains a particular challenge. A small motion can easily be mistaken as an inlier of the background camera motion and will become a source of error in the estimation of camera pose.

In general, motion segmentation is based on data driven bottom-up approaches like geometric, algebraic, model selection, flow based and tracking methods [7]. Most of these techniques use motion cues from texture-based optic flow and structure-based iterative closest point algorithms to obtain data correspondences. These methods require prior knowledge of

[1] School of Engineering, RMIT University, Victoria, Australia.
[2] School of Science, RMIT University, Victoria, Australia.
[3] School of Science, Edith Cowan University, Australia.
This work was supported by the Australian Research Council through an ARC Linkage Project grant (LP160100662).

the number of moving objects and work well only if the correct data correspondences are known.

With recent advances in deep learning, several methods have emerged that can generate accurate and reliable semantic segmentation [8]. This led to top-down approaches for motion segmentation based on tracking of objects detected by such high-level instance segmentation methods [9], [10]. Methods like Mask-RCNN [8] and RGB-D data-based segmentation [11] fail to detect objects when there are not enough 2D or 3D features. However, as we show here, this situation may be salvageable by using motion information. These learning methods also miss objects unseen in training data as the detector is only trained on a small range of object classes (black arrow in Fig 1 (c) for Mask-RCNN).

Combining high-level information from top-down approaches and low-level information from data-driven bottom-up approaches leads to many advantages. High-level information helps bottom-up approaches by providing object specific sampling and correspondence matching leading to accurate object specific motion estimation. Low-level information helps top-down approaches to correct the missing detection problem by using the motion cues to detect anything that moves. Our method segments unidentified objects due to the combination of both approaches (Fig 1 (d)).

The main issue of combining top-down and bottom-up approaches is the over segmentation of moving parts. All detected objects are tracked and motion models are generated for each one of the detected objects. Static objects (objects moving similar to the background) and dynamic objects are not differentiated (red arrow in Fig 1 (c)).

To resolve this issue, we propose to use statistical inference methods to combine similar motions (called statistical model fusion) and convert object-based segmentation (usually over-segmented) to accurate motion segmentation. An example of combining similar motions is shown in Fig 1 (d) (static object: table is combined with background). In this example, the background motion is made more detectable as a result of considering extra features associated with the static object (table). The use of statistical inference in this case leads to better accuracy for both motion parameter estimation and segmentation (also improves camera ego-motion or background motion estimation).

The main contribution of this work is to introduce a new framework, which combines top-down and bottom-up cues, to perform motion segmentation by formulating the problem as a statistical inference problem. The significance of this method is that it is capable of handling small motions of small objects and combining objects with coherent motion in the presence of high percentage of outliers (wrong data associations and failure in object detection). This is achieved by:

- A top-down approach that uses semantic information to perform object specific sampling: enabling concentration on small motions of small objects (in line with [9], [10]).
- A bottom-up approach that uses the object specific data from top-down approach to estimate motion parameters by measuring goodness of the data association, to switch between variants of Iterative Closest Point (not requiring data association) and Robust model fitting methods (requires data association).
- A new approach for combining similar motions by formulating the problem as a statistical inference, and solve the over segmentation issue by using one of the Extreme Value Theorem [12], Kolmogorov–Smirnov test [13] or Wilcoxon signed-rank test [14].

This paper is organized as follows: Section II introduces the related work. Section III provides description of our motion segmentation pipeline and the statistical model fusion algorithm. Section IV shows a comparative study of three different model fusion methods on a synthetic dataset. Ablation studies on real data shows performance with and without using model fusion. Experimental results on RGB-D Multi-Body motion segmentation dataset shows the effectiveness of our method to identify small motions. Section V concludes the paper and discusses future work.

II. Related Works

A. Motion Segmentation: Definition

Motion segmentation in a complex dynamic scene involves clustering areas with coherent motion, and estimating their motion parameters (which can include camera ego-motion). Grouping of parts or pixels that are moving with coherent motion is an ill-defined problem. For instance, [15] defines and then attempts to resolve the ambiguities in motion segmentation for cases in which only a part of an object moves. According to their definition, the entire object should be segmented even if a part of the object moves and objects remain stationary for a few frames need not be segmented (unlike tracking). In contrast to motion segmentation, video object segmentation involves the identification of objects of interest in the entire video. Those objects are considered the main foreground that are moving partially between frames. A detailed description of these two problems and a survey of their solutions are provided by [16].

B. Motion Segmentation: 2D methods

To provide a clear picture of the advantages of using different types of 2D motion segmentation methods, we classify those into distinct categories: geometric, algebraic, model selection, flow based and tracking methods. Those approaches are briefly described here.

Geometric methods segment data using epipolar constraints [17] while minimizing reprojection errors. Algebraic approaches listed include: Generalized Principal Component Analysis (GPCA) method [18], which uses feature trajectories and factorizes data into multiple subspaces, Subspace clustering, which represents high-dimensional data by the union of low-dimensional subspaces each associated with a separate motion, and Sparse Subspace Clustering (SSC) method [19], which was developed to handle missing correspondences in motion data by using sparse representation of motion clusters.

Model based methods select the motion models that best fit segmented data using RANSAC [20] or its variants [21], [22]. The models are based on Fundamental matrix, Essential matrix, Homography or Affine transformations between

consecutive images. Fitting procedures generally prefer higher number of model instances to increase fidelity to data (causing over-segmentation) and choose complex models (with higher degrees of freedom) to increase smoothness when the number and type of motion models are unknown a priori. Achieving an effective balance between fidelity and smoothness is shown to be difficult [23]. However, these methods have shown to be capable of handling degenerate motions.

Flow-based techniques rely on the brightness constancy assumption, and perform motion segmentation by either clustering of optic flow [24] or scene flow [25]. These methods often fail when there are degenerate motions or textureless objects. Tracking methods generally use particle filters [26] for estimating motion models. The tracking methods are limited to scenarios with only a few moving objects as tracking multiple objects increases the number of required particles exponentially.

C. Motion Segmentation: 3D methods

In contrast to 2D methods that are limited to using brightness patterns, 3D motion segmentation can take advantage of depth measurements from Kinect type sensors (RGB-D data) to achieve either sparse feature based or dense pointwise based motion segmentation. Sparse feature based methods, like the one proposed by Gruber et al. [27], track a set of feature points, to factorize noisy data into multiple moving objects. The issue is that factorization based methods are offline methods and require entire feature trajectories to perform segmentation. In another feature based method, proposed by Rothganger et al. [28], motion segmentation is performed by tracking and matching features of planar patches of a 3D scene. Consequently, 3D objects that cannot be modelled by a combination of planar patches are not tracked. Agrawal et al. [29] used matched features from stereo pairs to obtain 3D points, which were tracked to estimate 3D motion segmentation using RANSAC. This method cannot handle multiple moving objects. Samunda et al. [30] used the Delaunay triangulation method on distances between feature points to achieve segmentation. This method can therefore only work for rigid objects. Multimotion Visual Odometry [31] uses stereo data, applies multi model fitting on sparse features to create tracklets and improves the initial segmentation by merge and split operations.

Dense pointwise 3D segmentation methods are developed for scene reconstruction in SLAM and autonomous driving applications. Dense methods, like the one proposed by Roussos et al. [32], use RGB images to estimate dense motion segmentation and reconstruct 3D scenes by energy based multiple model fitting methods. In contrast, methods proposed by Stückler et al. [33] and Bertholet et al. [34] use RGB-D data. The former introduced a dense RGB-D motion segmentation using expectation–maximization (EM) on point clouds converted to 3D voxels at different resolutions while the latter proposed energy minimization by introducing a sensor noise model and an occlusion handling method to segment temporally consistent tracklets. The above methods use batch processing on entire sequence to perform segmentation.

D. High-level Semantic information-based motion segmentation

Advances in deep learning for object recognition have enabled the use of high level semantic based information in different computer vision applications. Motion segmentation is no exception and the advent of accurate and reliable semantic segmentation tools such as Faster-RCNN [35] or Mask-RCNN [8] has led to use of high-level interpretations for motion segmentation. For instance, the 2D motion segmentation method presented in [36] combines semantic and geometric properties in a way that high-level object motions of real objects like pedestrians are modelled as composition of many low-level rigid motions.

For RGB-D motion segmentation using semantic information, Co-Fusion [37] uses motion cues or instance segmentation to perform multiple model fitting and track the segmented objects using Elastic-Fusion [38] to produce surfel maps of the static environment. Mask-Fusion [9] extends Co-Fusion [37] by using semantic segmentation from Mask-RCNN [8] (refined with geometric edge information) to track individual objects and generate semantically labelled surfel maps. Such instance object segmentation based methods cannot handle non-rigid objects and their tracking would fail in cases where objects lack texture and are not identified by recognition tools. To take advantage of motion information, MID-Fusion [10] was proposed in which the Mask-RCNN [8] based object segmentation results are refined by both motion and geometric information. Volumetric Octree maps capable of modelling free space and connectivity were used for tracking every object, individually. EM-Fusion [39] has recently been proposed to add the occlusions handling to the above framework and uses signed distance functions to represent objects.

Similar to above methods [9], [10], [39] , our approach targets the use of high-level semantic information. However, existing methods are unable to combine static objects with static background (treat them as different objects) and their tracking is heavily dependent on the success of the object recognition part. In contrast, our system combines static objects with the static background to increase the detectability of the background motion (provide more features due to the inclusion of static objects) and tracks the dynamic objects, only. Our method also creates virtual masks for the unidentified objects using motion information and can therefore compensate for the object detection errors.

E. Dynamic SLAM

One of the important applications of motion segmentation is the Dynamic SLAM [7]. As real world applications often contain dynamic objects, the static environment assumption used in many SLAM methods, creates false feature correspondences. This leads to drift in pose estimation and false loop closures. Recent works address this issue by using motion segmentation to remove dynamic content from the data.

For instance, the well-known feature based ORB-SLAM2 [40] has been improved by being fed with only static content. In the method presented by Li and Lee [41], depth edge points (for every keyframe) were used to distinguish between

dynamic and static points by measuring their reprojection errors. To improve further, Zhang et al. [42] added line features to the depth edge features for estimating the likelihood of features being static. Similarly, Wang et al. [43] used a clustering method to segment trajectories and exclude dynamic points from the energy minimization step. This method would only work off-line as it requires long-term video trajectories. To take advantage of high level information, DynaSLAM [44] combines multi-view geometry and deep learning for tracking, mapping and inpainting in dynamic scenes while DS-SLAM [45] uses semantic information along with moving time consistency and tracking to remove dynamic objects. The use of high level information is shown to improve the accuracy of SLAM methods in dynamic environment.

The above feature-based SLAM methods perform poorly in the presence of motion blur, curved edges and low textured environments [7]. To improve the performance of dense RGB-D based SLAM in dynamic applications, the DVO SLAM [46] method was adapted in different ways. For instance, BaMVO [47] advocated learning for background modelling of depth data and using the model to reduce the influence of dynamic objects. Motion removal RGB-D SLAM [48] advocates building (and incrementally updating) a foreground model by learning and then inferring a pixel-wise dense likelihood (for being a foreground point). The paper also states that the method cannot handle small motions as reprojection errors of those small motions and the static objects, do not differ significantly.

III. Proposed Method

Given two consecutive color images I_{t-1} and I_t and the corresponding registered depth images D_{t-1} and D_t, the proposed motion segmentation algorithm outputs are:

- Number of independent moving objects K.
- Segmentation labels for each pixel $L = \{l_i\}_{i=1}^{N}$ where $l_i \in \{0, 1, 2, \cdots, K\}$ are independently moving objects with 0 representing the background and N is the number of pixels in the image.
- The motion Models for each independently moving object $F = \{F_i\}_{i=0}^{K}$

As the algorithm deals with 3D data, we use the six degrees of freedom rigid motion model represented by three-dimensional translation vector T and Euler angles of rotation forming a six tuple $F_m = \left\{T_x^{(m)}, T_y^{(m)}, T_z^{(m)}, R_x^{(m)}, R_y^{(m)}, R_z^{(m)}\right\}$. An overview of the proposed process flow pipeline with step by step results is shown in Fig 2. The first step is the top-down branch that performs semantic object segmentation. The next step is to take advantage of the high-level semantic segmentation and perform selective sampling to obtain point correspondences. The third one is the bottom-up part that uses data points from a specific sampled object to robustly estimate the motion model using data correspondences, or switching to Iterative Closest Point algorithm if appropriate correspondences are not found. The final step is to convert object based semantic segmentation to motion segmentation by combining similar motion models using a statistical inference theory.

The above pipeline is designed to use well established statistical model fusion theory to combine globally working high-level information (deep learning based semantic segmentation) to guide the locally working (data driven robust geometric model estimation method). We will show that the combined results are comparable to the best available methods while the proposed method can identify small moving objects that are frequently missed by other motion segmentation algorithms.

A. Object mask proposals

For the top-down approach, we feed color images I_{t-1} and I_t to Mask-RCNN [8] to obtain object masks. For each image we obtain J_M binary masks $\{M_j\}_{j=1}^{J_M}$ corresponding to J_M objects. A background mask is created as the collection of all pixels in an image not labelled by Mask-RCNN as an object: $M_0 = \overline{\{M_1 \cup M_2 \cup \cdots M_{J_M}\}}$.

For every mask M_j at time t, we calculate its corresponding mask $\widehat{M_j}$ at time $t-1$ using a simple thresholding on the percentage of overlapping area between them. This assumption holds as we are mainly dealing with small motions in videos that generally have high frame rates. In applications where the above assumption does not hold, one can use point correspondences from a method like DeepMatching [49] that is known to handle large motions.

Mask-RCNN may not detect some objects and this can cause false object correspondences. This problem is overcome by utilizing the motion cues provided by optic flow between I_{t-1} and I_t. First, the dominant motion in the background, F_0, is estimated using the background masks of the two frames I_{t-1} and I_t (M_0 and $\widehat{M_0}$) and the corresponding optic flow. Points (called outliers) in M_0 that do not belong to F_0 are identified using MSSE [50]. Those points are likely to belong to an undetected moving object if they are spatially grouped (see below). Based on this assumption, we cluster these points (outliers) into spatially consistent sub groups using the 3D Euclidean distance as the affinity measure:

$$A(i,j) = \|x_i - x_j\|_2^2 \tag{1}$$

where x_i, x_j are the 3D coordinates of each point. This leads to a simple clustering problem that can be solved in many different ways. Here, we use a simple iterative fit and remove procedure: chose a random point from outliers and calculate its distances to every other point using equation (1). The distances calculated are input to MSSE [50] to obtain the group of points that are spatially close to the chosen point. The selected points are then removed, and the above process is repeated until the number of remaining points are smaller than a fix threshold. This clustering procedure generates J_O binary masks $\{M_j\}_{j=1}^{J_O}$ corresponding to J_O undetected dynamic objects. The correspondences for background and all object masks are then updated using $J = J_M + J_O$ object masks.

B. Point correspondences and selective sampling

Optic flow is calculated by the publicly available median filtering based implementation of [51] to obtain 2D correspondences. We further improve the optic flow by imposing a constraint that restricts the matching between pixels in I_t

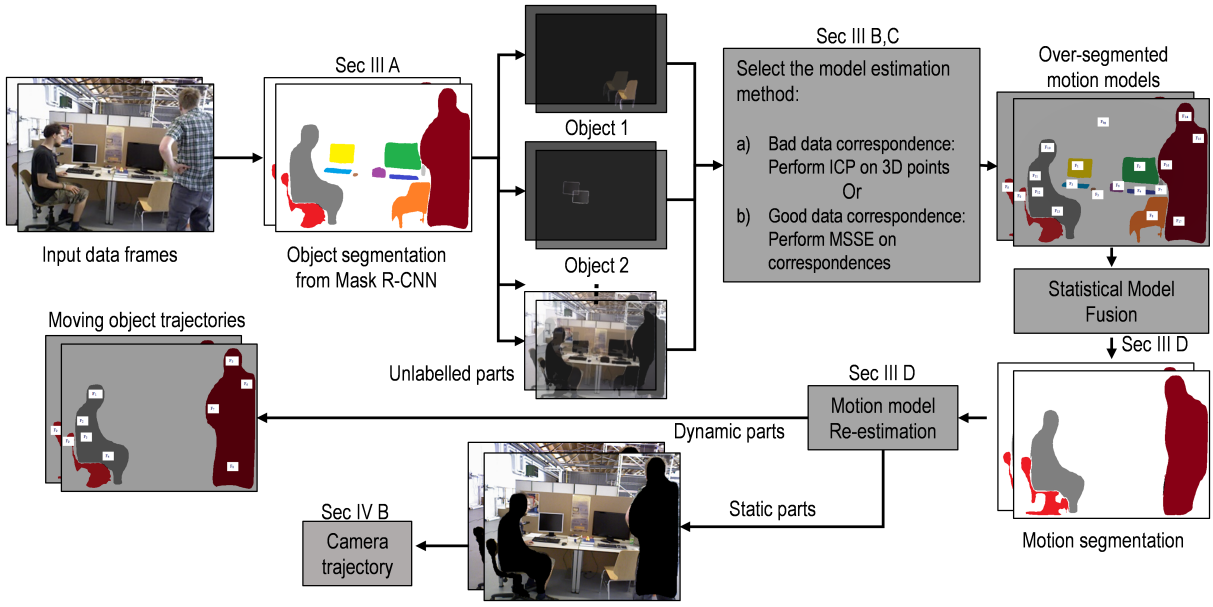


Fig. 2: The framework of our method. The raw images are converted to object segmentation using Mask-RCNN. Then the object-based sampling and matching are used to estimate initial object motion parameters. Resulting over-segmentation is resolved by combining similarly moving parts using statistical inference. Finally motion models are re-estimated and both camera and moving object trajectories are calculated.

belonging to object M_j and pixels that belong to the same corresponding object mask $\widehat{M_j}$ in I_{t-1}.

Many RGB-D algorithms typically use uniform downsampling to reduce the computational cost of their algorithms. However, such actions can lead to poor outcomes for small sized objects particularly when they are also far from the camera. To avoid this, we sample the point clouds X_t & X_{t-1} by using the size of object and its distance from camera, in a way that:

$$gridstep_i = \frac{k \cdot area\,(M_i)}{depth\,(M_i)} \tag{2}$$

where M_i is the mask for object i and k is a commonly used constant grid sampling factor (set to 5 in all of our experiments to keep the computation feasible).

C. Robust model estimation

In our proposed approach, initial estimates for the motion models are calculated for all the separately detected objects $F = \{F_j\}_{j=0}^{J}$ using the sampled point clouds X_t & X_{t-1} along with object masks $M = \{M_j\}_{j=0}^{J}$ and the improved optic flow field vector (u, v) (explained in the previous section).

Data correspondences of every object (or mask) is classified as good or bad based on the median color gradient within the object mask. Gradient magnitude is calculated using the Sobel operator with a 5-by-5 neighbourhood for the entire image I_t and the values are normalized between 0 and 1. The median (as a robust measure) value of gradient magnitudes for pixels belonging to the object mask, M_i, is then compared with a threshold (0.01 in all our experiments) to decide if the object has good data correspondences (with sufficient texture information).

Good data correspondences indicates that the object has sufficient texture and its motion model can be reliably estimated by registering X_t of its mask with their correspondences in X_{t-1} using 2D optic flow and depth information. Initial transformation R_{msse} and T_{msse} using the MSSE [50] algorithm is further refined by using a variant of Iterative Closest Point that includes color information (called CICP [52]) to obtain R_{icp} and T_{icp}. The total transformation is then presented by:

$$R = R_{icp} \cdot R_{msse}, \quad T = (R_{icp} \cdot T_{msse}) + T_{icp}. \tag{3}$$

The motion model for objects with bad data correspondences is estimated by registering X_t & X_{t-1} using ICP on depth values, only.

Articulated objects have to be considered differently as there won't be any single rigid transformation motion model that accurately explain their motions. Objects with low percentage of inliers with respect to the initial model estimates are also classified based on threshold as articulated objects for simplicity. We segment articulated objects with a single semantic label but represent them as a composite of multiple sub-segments with separate motion parameters (e.g. humans are represented by head, arms, legs and body each moving differently). This composite information is useful for applications like scene understanding, anomaly detection and action recognition [36].

For articulated objects classified by the above step, we perform spectral clustering to oversegment it into multiple rigid sub-segments based on their optic flow and spatial extent. The affinity matrix A for the clustering is defined as:

$$A\,(i,j) = \alpha \cdot A_m\,(i,j) + (1-\alpha) \cdot \|x_i - x_j\|_2^2. \tag{4}$$

where x_i, x_j are the 3D coordinates of the mask's points and A_m is the model based affinity matrix.

The first term A_m in equation (4) represents the motion coherence of two points for a given motion model and is calculated using Algorithm 1 of [22]. In this algorithm, a fixed number of motion model hypotheses (generated using random sampling of optic flow vectors within a mask) are combined to generate the affinities between all points within that mask. The second term in equation (4) represents the spatial smoothness and advocates for spatial contiguity. Combining these two factors results in articulated objects being segmented into multiple rigid moving parts, which paves the way for using rigid motion models.

The above procedure generates initial model estimates for three groups: objects with good or bad correspondences and articulated objects. This means that all objects, irrespective of their motions, are segmented and one needs to find a way to combine similarly moving (or stationary) objects.

D. Statistical model fusion

This paper presents an innovative statistical approach for converting initial object-based segmentation (and motion model estimates) $F_{in} = \{F_j\}_{j=0}^{J}$ to motion based segmentation $F_{out} = \{F_k\}_{k=0}^{K}$. The approach is based on posing the problem of identifying similar motions (e.g. combining static objects with background) as a statistical inference problem.

At the beginning, one object and its initial model estimate is chosen as a reference model (e.g. background M_0 and its motion model F_0). The reference model can then be characterized by a distribution (called reference distribution) formed using a normalized histogram of the reference object residuals with respect to its model. Residuals are taken to be the norm of Euclidean difference between points in X_t belonging to M_0 and transformed corresponding points in X_{t-1} using F_0. One would expect to see a χ^2 type distribution (with zero mode) for a properly segmented object.

To test if another object M_j has a motion similar to the reference object M_0, a test distribution is formed using the residuals of the new object with respect to the reference motion model F_0. We know that by definition, two objects with similar motions should have similar test and reference distributions while objects with different motions can't have the same (based on a statistical significance measure) test and reference distributions. Test distribution of a dynamic object should have a non-zero mode signifying the fact its motion is different from the reference model.

The art of deciding if two sets of observations are samples of the same distribution is called statistical hypothesis testing. There are two well-known test statistics, called Kolmogorov–Smirnov-Test (KS) and Wilcoxon-Signed-Rank-Test (WSR), to compare observations X from reference distribution and observations Y from test distribution. We have also adapted the statistical Extreme-Value Theory (EVT) to generate a test for the same problem. These three methods are explained here and their effectiveness on synthetic and read data are compared in section IV.

1) Kolmogorov–Smirnov test: KS-Test is a nonparametric statistical test [53] that uses an empirical cumulative distribution function to compare either one set of samples with

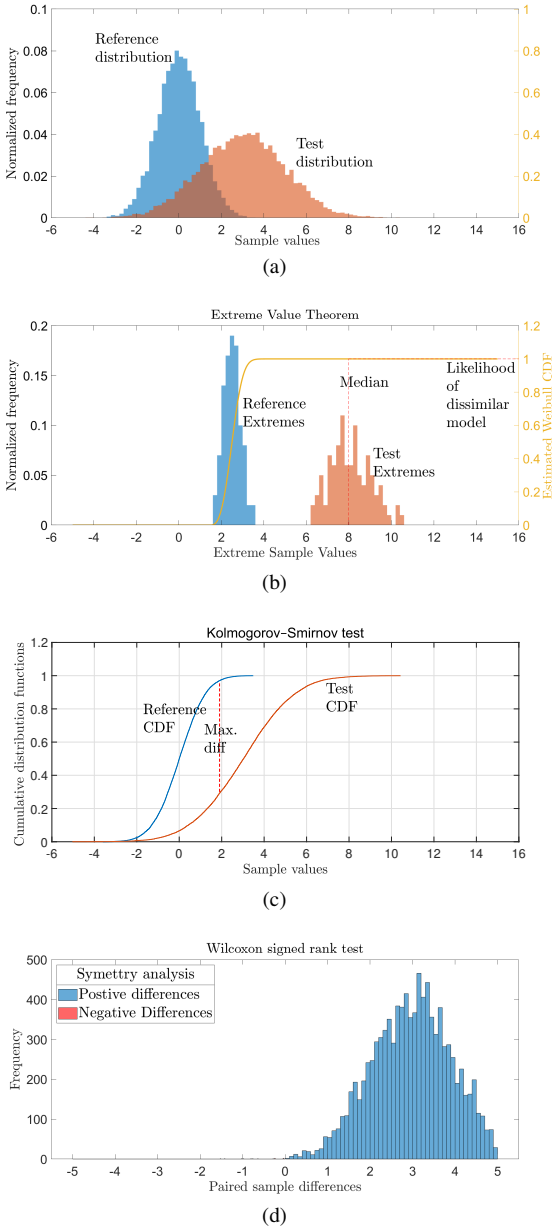


Fig. 3: Testing similarity of two different distributions using three Statistical methods. (a) Two input distributions. (b) Shows the estimated Weibul CDF of the reference and its associated value for the median likelihood estimate of the test distribution for EVT-Test. (c) Comparison of the empirical CDFs of reference and test distributions for KS-Test. (d) Symmetry analysis for WSR-Test.

Algorithm 1 Statistical model fusion.

Input: Point clouds X_t, correspondences in X_{t-1}, Object masks M and corresponding initial motion models F_{in}.
Output: Fused masks M and combined motion models F_{out}.

$F_{out} = \varnothing$
repeat
  $i \leftarrow$ index of a random reference model in F_{in}.
  $X_t^{(i)}, X_{t-1}^{(i)} \leftarrow$ Points in mask M_i.
  $X_{inliers}^{(i)} \leftarrow$ Inliers to model F_i.
  $E_i \leftarrow$ Ref. sample - Residuals of $X_{inliers}^{(i)}$ to F_i.
  for all model j in $F_{in}, j \neq i$ **do**
    $X_t^{(j)}, X_{t-1}^{(j)} \leftarrow$ Points in mask M_j.
    $X_{inliers}^{(j)} \leftarrow$ Inliers to model F_j.
    $E_j \leftarrow$ Test sample - Residuals of $X_{inliers}^{(j)}$ to F_i.
    $f \leftarrow Fuse(E_i, E_j, Method)$
    if $f = True$ **then**
      Fused mask $M_i \leftarrow M_i \cup M_j$
      $F_i \leftarrow$ Re-estimate model by new M_i using (3)
      $F_{in} = F_{in} \setminus F_j$, $M = M \setminus M_j$.
    end if
  end for
  $F_{out} = F_{out} \cup F_i$
  $F_{in} = F_{in} \setminus F_i$.
until $F_{in} = \varnothing$

a reference distribution or two sets of samples directly (two sample KS test). To perform the latter, empirical cumulative distribution functions are built from the two sets of given samples. Empirical cumulative distribution function (CDF) for n independently and identically distributed (iid) samples of set X is as follows.

$$F_X(x) = \frac{1}{n} \sum_{i=1}^{n} I_x(X_i) \tag{5}$$

where $I_x(X_i)$ is an indicator function such that:

$$I_x(X_i) = \begin{cases} 1 & \text{if } X_i \leq x \\ 0 & \text{if } X_i > x. \end{cases} \tag{6}$$

The maximum difference between two empirical CDFs provides a distance statistic.

$$D_{X,Y} = \sup_t |F_X(t) - F_Y(t)| \tag{7}$$

The above distance has in the past been used for adaptive clustering [54]. Here, we use this to fuse similar motion models (for a given confidence interval). In this test, $D_{X,Y} \leq t_\alpha$ indicates that there is not enough evidence to conclude that sets X and Y follow different distributions. However, $D_{X,Y} > t_\alpha$ indicates the existence of strong evidence for sets X and Y to be samples of different distributions. The value of t_α is calculated based on the chosen significance level α and the number of samples in sets X and Y [53].

2) Wilcoxon signed rank test: Another popular nonparametric statistical test is the WSR-Test [55]. The test is based on using rank of differences between two sets of samples to decide on their similarity. The test extends the student t-test, which only works for comparing normal distributions, to be applicable to compare samples of general nonparametric distributions. In this method, differences between paired observations from both sets are calculated and ranked based on their absolute values. The minimum (W) of the sums of the ranks of both positive (W^+) and negative (W^-) differences are then used to calculate the probability of samples having the same distributions (p). Two sets of X and Y are said to be sampled from the same distribution if the difference between their paired observations are symmetric around zero ($P(X > Y) = P(Y > X)$) and otherwise if the differences are non-symmetric around zero.

$$W^+ = \sum_{\forall i: sgn(X_i - Y_i) = +ve} rank\left(|X_i - Y_i|\right) \tag{8}$$

$$W^- = \sum_{\forall i: sgn(X_i - Y_i) < -ve} rank(|X_i - Y_i|) \tag{9}$$

$$W = min\left(W^+, W^-\right) \tag{10}$$

Similar to KS-Test, $p \geq \alpha$ is an indication of lack of evidence for sets of X and Y to be samples of different distributions. α is called the significance level and is commonly set as 0.05.

3) Extreme Value Theorem: EVT [12] has been adopted for modeling extreme events in weather and financial systems. The theorem states that: For $M_n = \max(X_1, \ldots, X_n)$, where $(X_1, X_2, \ldots)$ are a sequence of i.i.d. samples drawn from any distribution, if there exist a sequence of constants $\{a_n > 0\}$ and $\{b_n\}$ such that:

$$\lim_{n \to \infty} P\left(\frac{M_n - b_n}{a_n} \leq x\right) = G(x) \tag{11}$$

and G is a non-degenerate function, then G must belong to one of the following distribution families: *Gumbel*, *Fréchet* and *Weibull*.

The distribution G belongs to Weibull family if the distribution has limited tails. Given the residuals are bounded, in our proposed method, extreme values of observations X of reference distribution is characterized by a three parameter Weibull distribution. The shape k, location γ and scale λ parameters of the Weibull distribution are estimated by the Maximum Likelihood method. Extreme values from observations Y of test distribution are used to check the likelihood of two motions being similar by using the reference cumulative distribution in 12 as likelihood function. To combine motions, we compare the median likelihood of the extreme values of Y with a predefined threshold.

$$L(x \mid k, \gamma, \lambda) = 1 - e^{-\left(\frac{x-\gamma}{\lambda}\right)^k} \tag{12}$$

Figure (3) shows how each of the three mentioned test statistics differentiates between reference normal distribution $\mathcal{N}(\mu, \sigma^2)$ with $\mu = 0$ and $\sigma = 1$ and test normal distribution with $\mu = 3$ and $\sigma = 2$. In this example and for the sake of simplicity, Gaussian distributions are used but residuals usually form χ^2 like distributions, which will be discussed in Section IV-A.

Algorithm 2 Using the three test statistics.

Input: Reference set E_i, Test set E_j, Method to be used.
Output: $f = True$ if sets follow same model.
function $Fuse(E_i, E_j, Method)$
  if $Method = EVT$ **then**
    $Ext_i \leftarrow$ Max of l random samples from E_i for p times.
    $k, \lambda, \gamma \leftarrow$ Estimate Weibull parameters with Ext_i.
    $Ext_j \leftarrow$ Max of l random samples E_j for p times.
    Calculate likelihood L_j of median of Ext_j using (12).
    if average(L_j) $<$ Threshold **then**
      $f = True$
    end if
  end if
  if $Method = Kolmogorov\ Smirnov\ test$ **then**
    Calculate empirical $c.d.f\ F_X(t)$ from E_i using (5).
    Calculate empirical $c.d.f\ F_Y(t)$ from E_j using (5).
    Calculate distance function $D_{X,Y}$ using (7).
    if $D_{X,Y} \leq t_\alpha$ **then**
      $f = True$
    end if
  end if
  if $Method = Wilcoxon\ signed\ rank\ test$ **then**
    Calculate p and W from E_i and E_j using (8,9 and 10).
    if $p \geq \alpha$ **then**
      $f = True$
    end if
  end if
end function

Given the initial motion models F_{in}, we execute algorithm (1) to obtain the combined motion models F_{out}. In the above algorithm, statistical model fusion is used to check if two models (or masks) belong to the same motion. It they are from the same motion, we update the segmentation by joining their object mask proposals to get the fused object mask. Then, for the fused object mask, the motion model is re-estimated using the procedure given in Section III-C. (use the fused masks instead of the object-based mask proposals).

IV. Experimental Results

In the first part, we conducted a comparative study on a synthetic dataset to test the performance of the proposed model fusion using three different test statistics and identified their relative advantages. We performed an ablation study on real data (TUM-RGBD benchmark [56]) to demonstrate advantages of including the model fusion part to existing object instance segmentation methods for dynamic object removal in SLAM.

To show that our method performs accurate segmentation for small or slow moving objects, we applied our method to Multibody Motion Segmentation dataset [57] containing small, medium and large size objects and compared our results with existing methods. We also show comparative results for all sequences of the SBM-RGBD dataset [58] as those involve multiple moving objects.

We perform all experiments on a PC with intel corei7-7600U CPU with 2.80GHz and 16GB RAM. The average computation time of each segment of the proposed method is as follows: object mask proposal generation takes 273 ms/frame, the selective sampling / data association steps take 269 ms/frame, the robust model estimation takes 457 ms/frame and model fusion takes 28 ms/frame. On top of the above our algorithm takes segmentations from Mask-RCNN (300 ms/frame) and optic flow as inputs (deep learning based Flownet2 [59] takes around 7 ms/frame where as traditional methods like [51] takes 70 s/frame).

A. Comparing model fusion methods : Synthetic dataset

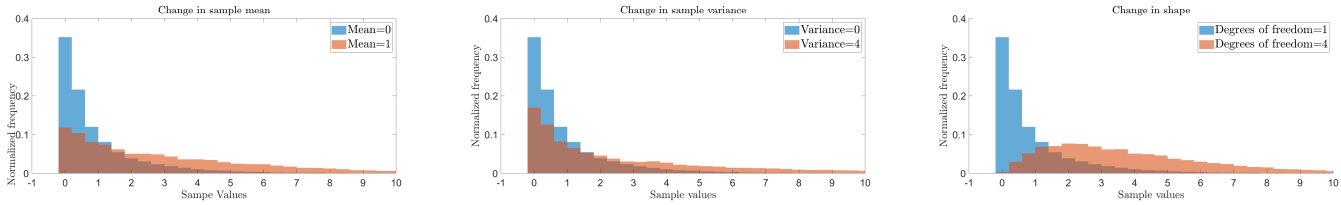
We generated numerous sets of observations by sampling from a χ^2 distribution with known parameters. The first sample is considered as reference χ^2 distribution. The second set also has the same parameters of the reference set but has different number of observations. As shown in Figure (4), the third set was generated by varying only one of the sample mean, variance or degrees of freedom. The second set was then used to test similarity and the third one was used to test dissimilarity. The median values of the F-Scores of these experiments are reported in Table (I) for 1000 iterations of the experiment.

1) Case1: Variation of sample mean: The performance of three tests for varying differences in sample means of test and reference distributions were examined. Even small differences between sample means can be detected by the Wilcoxon signed rank test as this test is an extension of student t-test and concentrates more on the central tendency (while the other two tests concentrate on the shape).

2) Case2: Variation of sample variance: Some distributions can have the same central tendency while they have different spreads. So, an ideal method needs to be sensitive to the changes in sample variance. WSR-Test is not effective for detecting the change of variance as it concentrates on changes in sample means. The KS and EVT tests however are sensitive to such changes.

3) Case3: Variation of shape: Some distributions can have the same medians, suggesting the same central tendency, but differ in the shape of their left and right tails. This particularly happens if the spread of the distributions is not symmetric. WSR-Test is again not effective for this, while the KS-Test performs reasonably well. EVT overall is more sensitive to shape changes as it is based on modelling of extreme events that are samples from tails of distributions.

The results of our simulations are shown in table I. These results show that the WSR-Test is more sensitive to changes in sample mean,while the KS-Test is more sensitive to changes in sample variance and the EVT-Test is more sensitive to changes in the tails of distributions. The EVT-Test requires less computation time. The results show that the WSR-Test is not robust if the tested distributions only differ in the variance or tails. As this test needs paired observations, the number of samples as well as the presence of zero difference or tied absolute differences affect its performance. Existence of outliers can affect the EVT-Test and its implementation needs a pre-processing step to eliminate noisy observations. It is also affected by the sample size as low sample size leads to more



(a) (b) (c)

Fig. 4: Visualization of two χ^2 distributions with different sample mean, variance and degree of freedom (three cases of the ablation study).

uncertainty in the estimation of the Weibull parameters. The KS-Test is a more general statistic for comparing any type of difference in the two distributions while it is less discerning than the WSR-Test for finding central tendency shift and the EVT-Test for finding tail changes. Overall, as there is no clear winner, one can use either of the KS-Test or EVT-Test to perform the statistical inference task.

TABLE I: F-Score accuracy for change in Sample Mean,Variance and shape

Variation	Values	F-Measure		
		Extreme Value Theorem	Kolmogorov Smirnov test	Wilcoxon signed rank test
Mean	3.00	0.9796	0.9848	**0.9899**
	1.00	**0.9796**	0.9744	0.9724
	0.30	0.6712	0.9118	**0.9146**
	0.10	0.6622	0.6458	**0.6667**
SD	2.00	0.9848	**0.9896**	0.3877
	1.75	0.9744	**0.9796**	0.4039
	1.50	0.9637	**0.9899**	0.3729
	1.25	0.9474	**0.9950**	0.4258
Degrees of freedom	2	0.9148	**0.9637**	0.6644
	3	**0.9899**	0.9691	0.6875
	4	**0.9796**	0.9583	0.7589

B. Ablation study: Real dataset -TUM-RGBD dynamic SLAM

To examine the significance of the proposed statistical model fusion step, we performed an ablation study and measured the improvement of our dynamic SLAM system on the TUM walking-xyz sequence with and without the model fusion part. The improvements are measured by both Absolute Trajectory Error (ATE), which is a measure of global consistency, and Relative Pose Error (RPE), which is a measure of translational and rotational trajectory drift errors of estimated camera pose.

We use the open-source DVO SLAM [1] for Camera Trajectory Estimation. It is a dense vSLAM method for RGB-D data that minimizes both photometric and depth loss. In addition, DVO-SLAM adaptively selects keyframes and uses them for loop closure. In our implementation we extract the background segmentation (Section III-D) and directly feed it to the DVO SLAM algorithm.

The performance was measured using root mean squared errors (RMSE) of both Absolute Trajectory Error (ATE) and Relative Pose Error (RPE). The above measures are calculated as follows [56]: For a given camera pose sequence $P_{1:n}$ (estimated trajectory) and $Q_{1:n}$ (ground truth trajectory) $\in SE(3)$, the Relative Pose Error (E_i) and Absolute Trajectory Error (F_i) are represented as:

$$E_i = (Q_i{}^{-1}Q_{i+\delta})^{-1}(P_i{}^{-1}P_{i+\delta}) \quad (13)$$

$$F_i = Q_i{}^{-1}\mathrm{SP}_i \quad (14)$$

where S is the transformation to align both trajectories and δ is the interval time.

Table II demonstrates the quantitative results. The first column shows errors for the original DVO SLAM [46] algorithm designed for static environments. The second column shows errors of our method without the model fusion part. This is obtained by removing all detected objects and integrating only the background to the front-end of the DVO SLAM.

The next three columns show the errors after the inclusion of the model fusion part, using the above three tests for joining static objects with the background. This leads to richer backgrounds containing more featured regions, which improves the camera ego-motion estimation. The improvement is calculated as:

$$I = \left(1 - \frac{w_{EVT} + w_{KS} + w_{WSR}}{3 \times \ w_0}\right) \times \ 100\% \quad (15)$$

where I represents improvement value, $((w_{EVT} + w_{KS} + w_{WSR})/3)$ and w_0 represent error values with and without model fusion part. Figure (5) shows ATE and RPE plots with and without model fusion part. The figure shows significant improvement in ego-motion estimation when using the statistical model fusion and including static objects in its calculation.

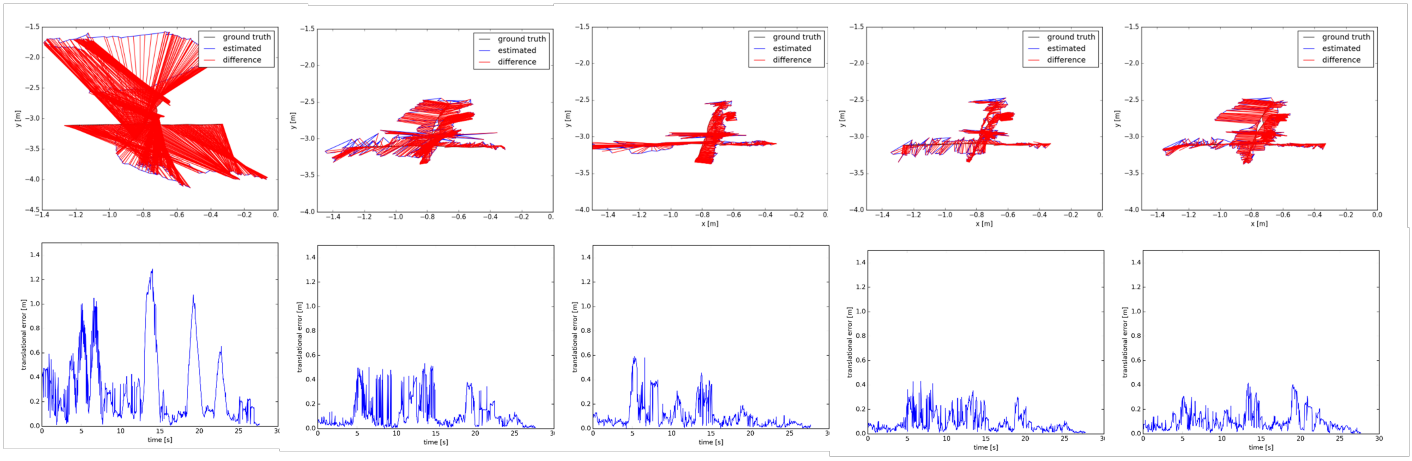
The performance of the proposed method is also compared with several related state-of-the-art approaches in Table III including: (1) Co-Fusion (CoF [37]) models and tracks multiple objects using motion cues and improving them over time through fusion in dynamic scenes. (2) BaMVO [47] that builds background model from depth data to reduce influence of dynamic objects. (3) Mask-Fusion (MF [9]) extends Co-Fusion [37] by using semantic information from Mask-RCNN [8] to track individual objects. (4) Motion removal DVO SLAM (MrDVO [60]) that builds and incrementally updates a foreground model and infers a pixel-wisely dense likelihood of being the foreground.

Although table III shows that the MrDVO [60] often produces lower errors, our method also performs segmentation

[1]https://vision.in.tum.de/data/software/dvo

TABLE II: RMSE of Absolute Trajectory Error (ATE) in m, Translational Drift (RPE) in m/s and Rotational Drift (RPE) in °/s without and with model fusion module of our approach.

Errors	DVO [46]	DVO + Ours without Model-Fusion	DVO + Ours with Model-Fusion			Improvement due to Model-Fusion(%)
			Extreme Value Theorem	Kolmogorov Smirnov test	Wilcoxon signed rank test	
ATE (m)	0.5966	0.1768	0.1383	**0.0942**	0.1037	36.65
Translational RPE (m/s)	0.4360	0.1805	**0.1209**	0.1211	0.1364	30.12
Rotational RPE (°/s)	7.6669	3.4638	3.1405	**2.6521**	3.0393	15.01



(a) (b) (c) (d) (e)

Fig. 5: ATE (top) and RPE (bottom) for fr3/walking-xyz from a) DVO SLAM, b) DVO + our method without model-fusion, DVO + our method with c) EVT model-fusion d) KS model-fusion and e) WSR model-fusion

TABLE III: RMSE of Translational Drift (RPE) in m/s and Rotational Drift (RPE) in °/s for dynamic sequences in TUM RGBD-Dataset.

Dataset	Translational RMSE (m/s)					Rotational RMSE (°/s)				
	CoF [37]	BaMVO [47]	MF [9]	MrDVO [60]	Ours	CoF [37]	BaMVO [47]	MF [9]	MrDVO [60]	Ours
fr3/sitting-stat	**0.0110**	0.0248	0.0170	-	0.0131	0.4400	0.6997	0.4300	-	**0.3458**
fr3/sitting-xyz	**0.0270**	0.0482	0.0460	0.0357	0.0393	**1.0000**	1.3885	1.2500	1.0362	1.0111
fr3/sitting-rpy	-	0.1872	-	-	**0.0856**	-	5.9834	-	-	**2.5162**
fr3/sitting-hs	**0.0300**	0.0589	0.0410	0.0547	0.0482	**1.9200**	2.8804	2.0700	2.2677	2.4978
fr3/walking-stat	0.2240	0.1339	0.0390	**0.0307**	0.0717	4.0100	2.0833	**0.0760**	0.8998	1.3333
fr3/walking-xyz	0.3290	0.2326	0.0970	**0.0668**	0.1209	5.5500	4.3911	2.0000	**1.5950**	3.1405
fr3/walking-rpy	-	0.3584	-	**0.0968**	0.1734	-	6.3398	-	**2.5936**	3.2944
fr3/walking-hs	0.4000	0.1738	0.0930	**0.0611**	0.1953	13.020	4.2863	3.3500	**1.9004**	4.9336

and tracking of each dynamic object in addition to ego-motion estimation similar to CoF [37] and MF [9]. Table III also shows that our method is accurate than CoF [37] and MF [9] in low dynamic sitting sequences due to inclusion of static objects with background using statistical inference. The assumption in MF [9] to treat all objects not being touched by a human as static object does not hold in real word applications.

C. Multibody motion segmentation dataset

We use our method to perform motion segmentation on three RGB-D video sequences from [57]. Each sequence consists of two rigid objects and non-rigid human hand moving small (cereal box and teacup), medium (watering can and box) and large (chairs) size objects. Partial ground truth is provided every five seconds. For comparison, we provide segmentation accuracy from [57] and [61] along with our results in table IV. The method presented in [61] works for small camera motions and as such, results are only reported for part of sequences (excluding frames with large camera motions) while the other competing method [57] works only for rigid objects and annotates ground truth of non-rigid motions by "dont-care" labels.

Using high-level information results has two advantages. First, our method improves the segmentation of small size objects by sampling selectively within an untextured and noisy background. Segmentation accuracy of large size objects (chairs) is also improved as using high-level information overcomes the noisy low-level motion information due to the rotationally repetitive arrangement of chairs' feet.

Fig (6) shows qualitative results for segmenting small, medium and large objects. Middle row shows failure of Mask-RCNN to detect the box on the table. As our proposed method

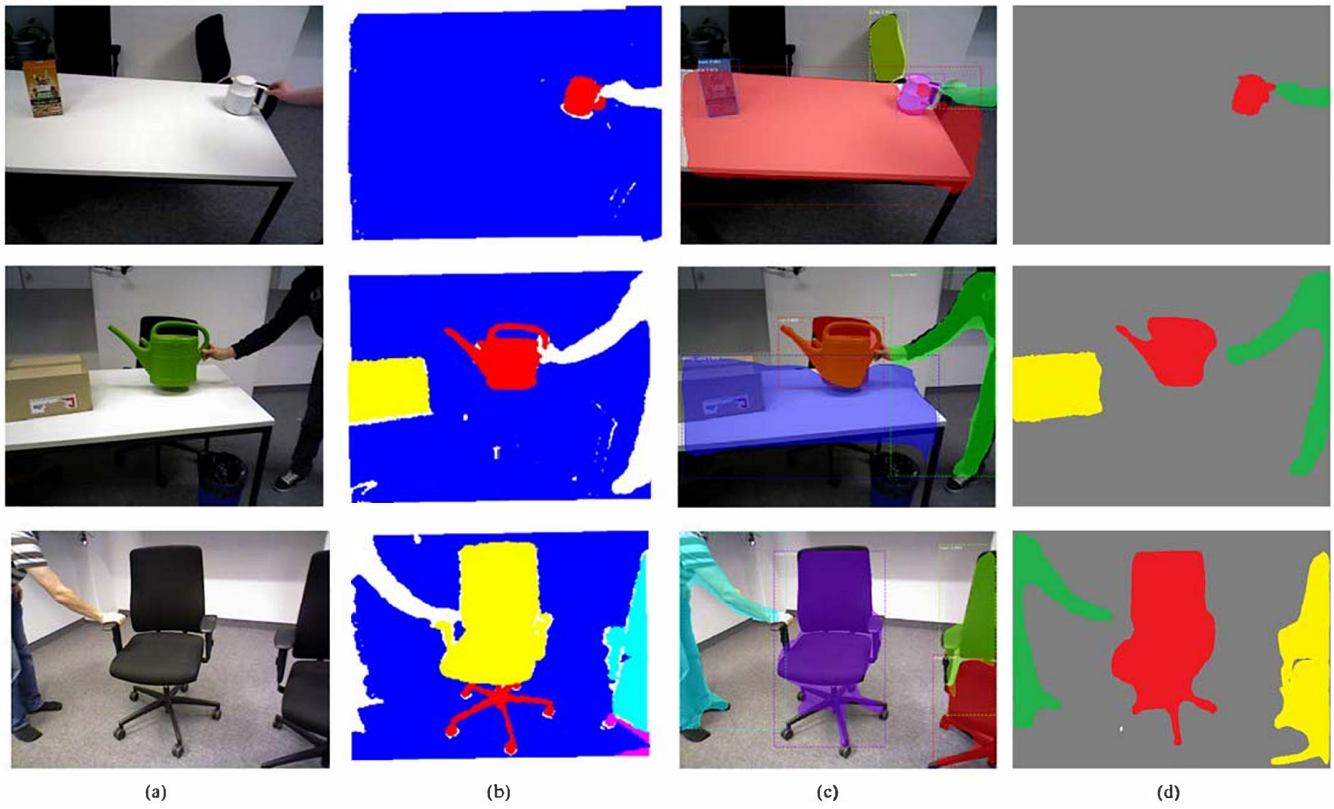


Fig. 6: Segmentation results for Small, Medium and Large object sequences. Column (a) Sample image. (b) Its ground truth segmentation. (c) Mask-RCNN object segmentation result. (d) Motion segmentation output of our method.

uses motion cues to create virtual mask, it can overcome this issue and the box is properly segmented. In the first two rows, the table, which is segmented by the Mask-RCNN, covers most of the scene and if it is removed, the background would not have much texture or structure for camera pose estimation. Again, our proposed method has been able to fuse the table with the background (no object area) and produced highly accurate segmentation.

TABLE IV: Segmentation accuracy for Small,Medium and Large object sequences

Size	Object	Segmentation Accuracy		
		From [61]	From [57]	Our Method
Small	Cup	0.9119	0.9500	**0.9708**
	Cerial Box	0.8719		**0.9754**
Medium	Watering can	0.9494	0.9400	**0.9857**
	Box	0.8469		**0.9876**
Large	Right Chair	0.7420	0.6300	**0.9828**
	Left Chair	0.7761		**0.9618**

D. SBM-RGBD dataset

SBM-RGBD dataset [58] includes 33 RGB-D videos acquired by the Kinect. There are 7 categories of videos for handling different challenges like Illumination Changes, Color Camouflage, Depth Camouflage, Intermittent Motion, Out of Sensor Range, Shadows and Bootstrapping. We use 5 sequences in the Out of sensor Range as it is the only category that contain multiple moving objects. It should be noted that the above dataset is aimed at evaluating background segmentation methods and the camera is kept static in all the sequences in SBM-RGBD dataset. Performance metrics reported for background segmentation are obtained by submitting the output segmentation from our method to the SBM-RGBD Challenge. The ground truth for motion segmentation were generated by manually annotating the random frames in each sequence for with the background segmentation's are available publicly (1080 frames out of 15000 frames in the test sequences). The performance of the proposed method is compared with several related state-of-the-art approaches (we have selected the top three methods in the SBM-RGBD challenge) including: (1) SCAD [63] that uses background subtraction based on both color and depth and combines them using graph cuts. (2) MFCN [62] that uses deep features learned from a multi-scale fully convolutional network to classify foreground and background. (3) RGBD-SOBS [64] that builds two neural background models using color and depth information and combines both outputs to segment the background. The first three methods can only provide background/foreground segmentation and work only in the presence of a static camera. Co-fusion and our method can both handle moving cameras in dynamic scenes. Table V compares the methods using average Precision, Recall and F-measures. The first three columns report the foreground/background segmentation metrics. The best results are obtained by [62]. This method uses a CNN

TABLE V: Foreground/Background Segmentation and motion segmentation F-Scores for the five sequences in Out of sensor Range category of SBM-RGBD dataset.

Method	Foreground/Background Segmentation			Motion Segmentation			
	Recall	Precision	F-Score	Recall	Precision	F-Score	IoU
MFCN [62]	**0.9917**	**0.9613**	**0.9763**	-	-	-	-
SCAD [63]	0.9286	0.9357	0.9309	-	-	-	-
RGBD SOBS [64]	0.9170	0.9362	0.9250	-	-	-	-
Co-Fusion [37]	-	-	-	0.4774	0.4945	0.4694	0.4213
Our Method	0.9420	0.9137	0.9264	**0.8587**	**0.6842**	**0.7418**	**0.7572**

based model to learn the background and uses that in the subsequent steps, therefore limiting it to static cameras. The other two top performing methods and our method show comparable accuracy. The next three columns show motion segmentation accuracy for our method and co-Fusion (Results generated by running the publicly available code[2] with the default parameters). The results show that our method has out-performed Co-Fusion in this dataset. This may be because SBM-RGBD dataset contains objects with both large and small motions (humans walking slowly and running).

Figure (7) shows qualitative results for foreground background and motion segmentation of our method along with ground truths for a random time frame in sequences: MultiPeople1, MultiPeople2 and TopViewLab3.

V. Conclusion

This paper proposes to use both high-level semantic information and low-level object specific motion cues, for performing motion segmentation of RGB-D data in complex dynamic scenes. The benefit of using instance object segmentation, along with motion cues, is to find small or slow moving objects among noisy background camera motion. Another benefit is the ability to overcome the problem of segmenting or tracking undetected moving objects (failures in object detection). Instead of finding seperate motions for each object, we use an innovative statistical inference method to combine static objects with the background and improve the accuracy of the camera pose estimation. Our method characterises each motion by a distribution using residual errors of object segmentation with respect to its motion models. Two motions with similar distributions are then fused using a statistical inference method. Experiments on synthetic data showed that KS-Test or EVT-Test are sensitive to changes in mean, variance or shape of two compared distributions while experiments on TUM-RGBD Dynamic SLAM dataset showed improvement in the accuracy of SLAM due to inclusion of the above model fusion part.

A limitation of our method is that it ceases to track moving objects that remain stationary for a few frames. This is because our method only uses information from two consecutive frames and does not use any long-term information. In the future work, we will explore the use of tracking based temporal consistency to address this issue.

[2]https://github.com/martinruenz/co-fusion

References

[1] A. Amiranashvili, A. Dosovitskiy, V. Koltun, and T. Brox, "Motion perception in reinforcement learning with dynamic objects," in *Conference on Robot Learning*, 2019, Conference Proceedings, pp. 156–168. 1

[2] K. Zampogiannis, K. Ganguly, C. Fermuller, and Y. Aloimonos, "Extracting contact and motion from manipulation videos," *arXiv preprint arXiv:1807.04870*, 2018. 1

[3] M. Siam, H. Mahgoub, M. Zahran, S. Yogamani, M. Jagersand, and A. El-Sallab, "Modnet: Motion and appearance based moving object detection network for autonomous driving," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, Conference Proceedings, pp. 2859–2864. 1

[4] J. H. Hammer, M. Voit, and J. Beyerer, "Motion segmentation and appearance change detection based 2d hand tracking," in *Information Fusion (FUSION), 2016 19th International Conference on*. IEEE, 2016, Conference Proceedings, pp. 1743–1750. 1

[5] M. Menze, C. Heipke, and A. Geiger, "Object scene flow," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 140, pp. 60–76, 2018. 1

[6] F. Fraundorfer and D. Scaramuzza, "Visual odometry: Matching, robustness, optimization, and applications," *IEEE Robotics and Automation Magazine*, vol. 19, no. 2, pp. 78–90, 2012. 1

[7] M. R. U. Saputra, A. Markham, and N. Trigoni, "Visual slam and structure from motion in dynamic environments: A survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 2, p. 37, 2018. 1, 3, 4

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2017, Conference Proceedings, pp. 2980–2988. 2, 3, 4, 9

[9] M. Runz, M. Buffier, and L. Agapito, "Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects," in *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2018, Conference Proceedings, pp. 10–20. 2, 3, 9, 10

[10] B. Xu, W. Li, D. Tzoumanikas, M. Bloesch, A. Davison, and S. Leutenegger, "Mid-fusion: Octree-based object-level multi-instance dynamic slam," *arXiv preprint arXiv:1812.07976*, 2018. 2, 3

[11] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from rgb-d images for object detection and segmentation," in *European conference on computer vision*. Springer, 2014, pp. 345–360. 2

[12] S. Coles, J. Bawa, L. Trenner, and P. Dorazio, *An introduction to statistical modeling of extreme values*. Springer, 2001, vol. 208. 2, 7

[13] F. J. Massey Jr, "The kolmogorov-smirnov test for goodness of fit," *Journal of the American statistical Association*, vol. 46, no. 253, pp. 68–78, 1951. 2

[14] J. D. Gibbons and S. Chakraborti, *Nonparametric statistical inference*. Springer, 2011. 2

[15] P. Bideau and E. Learned-Miller, "A detailed rubric for motion segmentation," *arXiv preprint arXiv:1610.10033*, 2016. 2

[16] R. Yao, G. Lin, S. Xia, J. Zhao, and Y. Zhou, "Video object segmentation and tracking: A survey," *arXiv preprint arXiv:1904.09172*, 2019. 2

[17] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003. 2

[18] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (gpca)," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 12, pp. 1945–1959, 2005. 2

[19] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, Conference Proceedings, pp. 2790–2797. 2

[20] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981. 2

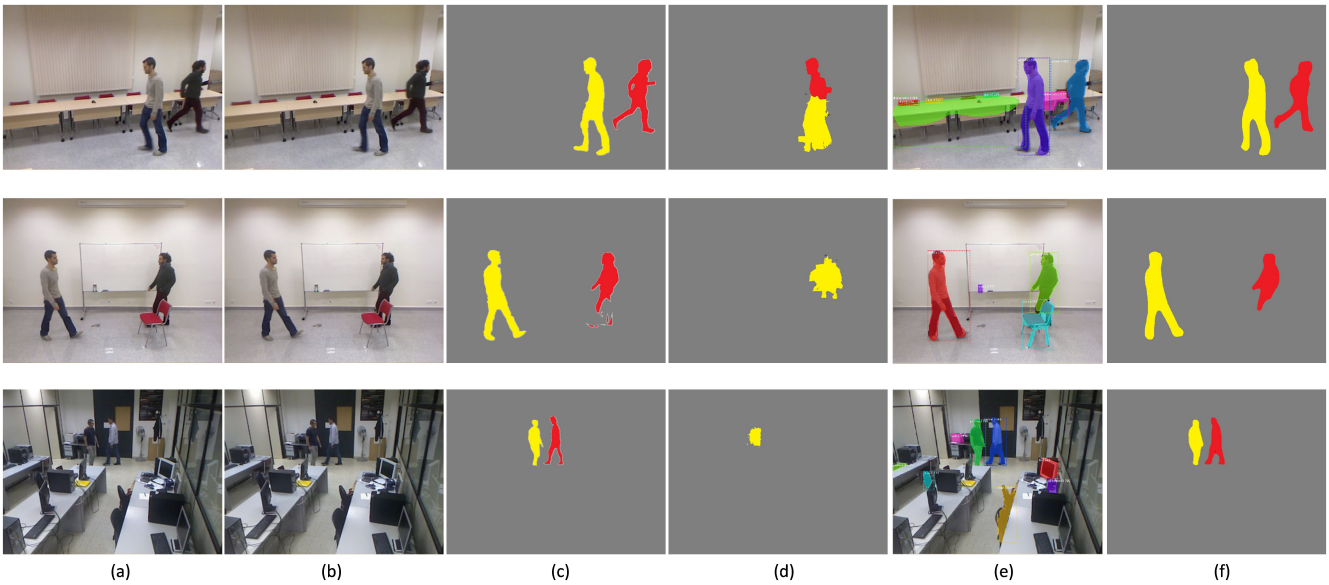


Fig. 7: SBM RGB-D segmentation results for 3 sequences MultiPeople1, MultiPeople2 and TopViewLab3. Columns (a,b) Consecutive image frames. (c) Ground truth segmentation. (d) Co-Fusion [37] results. (e) Mask-RCNN object segmentation result. (f) Motion segmentation by our method.

[21] R. B. Tennakoon, A. Bab-Hadiashar, Z. Cao, R. Hoseinnezhad, and D. Suter, "Robust model fitting using higher than minimal subset sampling," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 350–362, 2016. 2

[22] R. Tennakoon, A. Sadri, R. Hoseinnezhad, and A. Bab-Hadiashar, "Effective sampling: Fast segmentation using robust geometric model fitting," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4182–4194, 2018. 2, 6

[23] N. Gheissari and A. Bab-Hadiashar, "A comparative study of model selection criteria for computer vision applications," *Image and Vision Computing*, vol. 26, no. 12, pp. 1636–1649, 2008. 3

[24] K. Fragkiadaki, P. Arbelaez, P. Felsen, and J. Malik, "Learning to segment moving objects in videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, Conference Proceedings, pp. 4083–4090. 3

[25] P. F. Alcantarilla, J. J. Yebes, J. Almazán, and L. M. Bergasa, "On combining visual slam and dense scene flow to increase the robustness of localization and mapping in dynamic environments," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, Conference Proceedings, pp. 1290–1297. 3

[26] A. Kundu, K. M. Krishna, and C. Jawahar, "Realtime multibody visual slam with a smoothly moving monocular camera," in *2011 International Conference on Computer Vision*. IEEE, 2011, Conference Proceedings, pp. 2080–2087. 3

[27] A. Gruber and Y. Weiss, "Multibody factorization with uncertainty and missing data using the em algorithm," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1. IEEE, 2004, Conference Proceedings, pp. I–I. 3

[28] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce, "Segmenting, modeling, and matching video clips containing multiple moving objects," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 3, pp. 477–491, 2007. 3

[29] M. Agrawal, K. Konolige, and L. Iocchi, "Real-time detection of independent motion using stereo," in *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05)-Volume 1*, vol. 2. IEEE, 2005, Conference Proceedings, pp. 207–214. 3

[30] S. Perera and N. Barnes, "A simple and practical solution to the rigid body motion segmentation problem using a rgb-d camera," in *2011 International Conference on Digital Image Computing: Techniques and Applications*. IEEE, 2011, Conference Proceedings, pp. 494–500. 3

[31] K. M. Judd, J. D. Gammell, and P. Newman, "Multimotion visual odometry (mvo): Simultaneous estimation of camera and third-party motions," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, Conference Proceedings, pp. 3949–3956. 3

[32] A. Roussos, C. Russell, R. Garg, and L. Agapito, "Dense multibody motion estimation and reconstruction from a handheld camera," in *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 2012, Conference Proceedings, pp. 31–40. 3

[33] J. Stückler and S. Behnke, "Efficient dense 3d rigid-body motion segmentation in rgb-d video," in *BMVC*, 2013, Conference Proceedings. 3

[34] P. Bertholet, A. E. Ichim, and M. Zwicker, "Temporally consistent motion segmentation from rgb-d video," in *Computer Graphics Forum*, vol. 37. Wiley Online Library, 2018, Conference Proceedings, pp. 118–134. 3

[35] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, Conference Proceedings, pp. 91–99. 3

[36] P. Bideau, A. RoyChowdhury, R. R. Menon, and E. Learned-Miller, "The best of both worlds: combining cnns and geometric constraints for hierarchical motion segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, Conference Proceedings, pp. 508–517. 3, 5

[37] M. Rünz and L. Agapito, "Co-fusion: Real-time segmentation, tracking and fusion of multiple objects," *arXiv preprint arXiv:1706.06629*, 2017. 3, 9, 10, 12, 13

[38] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "Elasticfusion: Real-time dense slam and light source estimation," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1697–1716, 2016. 3

[39] M. Strecke and J. Stückler, "Em-fusion: Dynamic object-level slam with probabilistic data association," *arXiv preprint arXiv:1904.11781*, 2019. 3

[40] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017. 3

[41] S. Li and D. Lee, "Rgb-d slam in dynamic environments using static point weighting," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2263–2270, 2017. 3

[42] H. Zhang, Z. Fang, and G. Yang, "Rgb-d simultaneous localization and mapping based on combination of static point and line features in dynamic environments," *Journal of Electronic Imaging*, vol. 27, no. 5, p. 053007, 2018. 4

[43] Y. Wang and S. Huang, "Towards dense moving object segmentation based robust dense rgb-d slam in dynamic scenarios," in *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*. IEEE, 2014, Conference Proceedings, pp. 1841–1846. 4

[44] B. Bescós, J. M. Fácil, J. Civera, and J. Neira, "Dynslam: Tracking, mapping and inpainting in dynamic scenes," *arXiv preprint arXiv:1806.05620*, 2018. 4

[45] C. Yu, Z. Liu, X. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei, "Ds-slam: A semantic visual slam towards dynamic environments," *arXiv preprint arXiv:1809.08379*, 2018. 4

[46] C. Kerl, J. Sturm, and D. Cremers, "Dense visual slam for rgb-d cameras," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, Conference Proceedings, pp. 2100–2106. 4, 9, 10

[47] D.-H. Kim and J.-H. Kim, "Effective background model-based rgb-d dense visual odometry in a dynamic environment," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1565–1573, 2016. 4, 9, 10

[48] Y. Sun, M. Liu, and M. Q.-H. Meng, "Improving rgb-d slam in dynamic environments: A motion removal approach," *Robotics and Autonomous Systems*, vol. 89, pp. 110–122, 2017. 4

[49] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "Deepmatching: Hierarchical deformable dense matching," *International Journal of Computer Vision*, vol. 120, no. 3, pp. 300–323, 2016. 4

[50] A. Bab-Hadiashar and D. Suter, "Robust segmentation of visual data using ranked unbiased scale estimate," *Robotica*, vol. 17, no. 6, pp. 649–660, 1999. 4, 5

[51] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, Conference Proceedings, pp. 2432–2439. 4, 8

[52] M. Korn, M. Holzkothen, and J. Pauli, "Color supported generalized-icp," in *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, vol. 3. IEEE, 2014, pp. 592–599. 5

[53] F. J. Massey Jr, "The kolmogorov-smirnov test for goodness of fit," *Journal of the American statistical Association*, vol. 46, no. 253, pp. 68–78, 1951. 6, 7

[54] L. Mora-López and J. Mora, "An adaptive algorithm for clustering cumulative probability distribution functions using the kolmogorov–smirnov two-sample test," *Expert Systems with Applications*, vol. 42, no. 8, pp. 4016–4021, 2015. 7

[55] R. Shier, "Statistics: 2.2 the wilcoxon signed rank sum test," *Mathematics Learning Support Centre. Retrieved from http://www. statstutor. ac. uk/resources/uploaded/wilcoxonsignedranktest. pdf*, 2004. 7

[56] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Proc. of the International Conference on Intelligent Robot Systems (IROS)*, Oct. 2012. 8, 9

[57] J. Stückler and S. Behnke, "Efficient dense rigid-body motion segmentation and estimation in rgb-d video," *International Journal of Computer Vision*, vol. 113, no. 3, pp. 233–245, 2015. 8, 10, 11

[58] M. Camplani, L. Maddalena, G. M. Alcover, A. Petrosino, and L. Salgado, "A benchmarking framework for background subtraction in rgbd videos," in *International Conference on Image Analysis and Processing*. Springer, 2017, Conference Proceedings, pp. 219–229. 8, 11

[59] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2462–2470. 8

[60] Y. Sun, M. Liu, and M. Q.-H. Meng, "Motion removal for reliable rgb-d slam in dynamic environments," *Robotics and Autonomous Systems*, vol. 108, pp. 115–128, 2018. 9, 10

[61] Y. Kim, H. Lim, S. C. Ahn, and A. Kim, "Simultaneous segmentation, estimation and analysis of articulated motion from dense point cloud sequence," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1085–1092. 10, 11

[62] D. Zeng and M. Zhu, "Background subtraction using multiscale fully convolutional network," *IEEE Access*, vol. 6, pp. 16 010–16 021, 2018. 11, 12

[63] T. Minematsu, A. Shimada, H. Uchiyama, and R.-i. Taniguchi, "Simple combination of appearance and depth for foreground segmentation," in *New Trends in Image Analysis and Processing – ICIAP 2017*, S. Battiato, G. M. Farinella, M. Leo, and G. Gallo, Eds. Cham: Springer International Publishing, 2017, pp. 266–277. 11, 12

[64] L. Maddalena and A. Petrosino, "Exploiting color and depth for background subtraction," in *New Trends in Image Analysis and Processing – ICIAP 2017*, S. Battiato, G. M. Farinella, M. Leo, and G. Gallo, Eds. Cham: Springer International Publishing, 2017, pp. 254–265. 11, 12