ARTICLE IN PRESS

J. Vis. Commun. Image R. xxx (2013) xxx–xxx

Contents lists available at SciVerse ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Incremental object learning and robust tracking of multiple objects from RGB-D point set data

Seongyong Koo [a], Dongheui Lee [b], Dong-Soo Kwon [a,*]

[a] Human-Robot Interaction Research Center, Department of Mechanical Engineering, KAIST, Deajeon, Republic of Korea
[b] Department of Electrical Engineering and Information Technology, Technical University of Munich, 80290 Munich, Germany

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Multiple objects tracking
RGB-D point set data
Gaussian mixture models
3-*d* point set registration
Incremental learning
Robot vision
Visual tracking
Tempo-spatial data association

ABSTRACT

In this paper, we propose a novel model-free approach for tracking multiple objects from RGB-D point set data. This study aims to achieve the robust tracking of arbitrary objects against dynamic interaction cases in real-time. In order to represent an object without prior knowledge, the probability density of each object is represented by Gaussian mixture models (GMM) with a tempo-spatial topological graph (TSTG). A flexible object model is incrementally updated in the pro-posed tracking framework, where each RGB-D point is identified to be involved in each object at each time step. Furthermore, the proposed method allows the creation of robust temporal associations among multiple updated objects during *split*, *complete occlusion*, *partial occlusion*, and *multiple contacts* dynamic interaction cases. The performance of the method was examined in terms of the tracking accuracy and computational efficiency by various experiments, achieving over 97% accuracy with five frames per second computation time. The limitations of the method were also empirically investigated in terms of the size of the points and the movement speed of objects.

1. Introduction

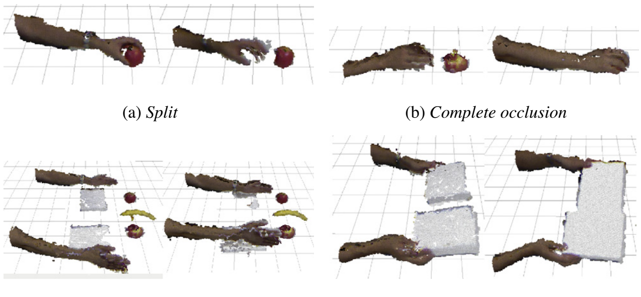
Identifying and tracking multiple moving objects from visual information is an ongoing challenge in many autonomous systems. With the advent of RGB-D cameras and improvements to point cloud data processing technologies [33], the observed environment can be represented as point set data $\mathscr{P} = \{p_1, \ldots, p_n\}$, wherein each point contains not only the RGB color but also 3-*d* position information, $p_i \in \mathbb{R}^6$. In particular, robust tracking of multiple objects from the RGB-D point data is necessary for service robots operating in human environments in order to reconstruct 3-*d* indoor environments [6], understand the semantic information of a human environment [6], and organize cluttered objects [7,21].

Especially in cases of learning complex actions manipulating multiple objects from human demonstrations [1,9,12,28], the tracking procedure presents difficult problems due to the lack of pre-knowledge of the objects and their movements. First, there are many flexibilities in the task. An object can be flexible and have articulated parts such as a human hand. The number of objects to track can also be flexible, because an object can newly appear in the scene or disappear from the scene. Second, dynamic movements of multiple objects cause various interaction cases between objects. Without object models, these situations distort the observed point data of each object, thus reducing the robustness of the tracking performance. For example, an element that was recognized as a single object can be separated into two individual objects, as in Fig. 1(a), and an object can be occluded by another object completely or partially, as in Fig. 1(b) and (c). In addition, multiple objects can be recognized as a single object when they are adjoining each other, as in Fig. 1(d).

The multiple object tracking problem of point set data involves identifying each point data, $\mathscr{P}^t = \{p_1^t, \ldots, p_n^t\}$, to each true object track, $\mathscr{T}^t = \{\{p_1^t, t_1^t\}, \ldots, \{p_n^t, t_n^t\}\}$, at each time. In order to solve this problem without any prior knowledge, this paper proposes a framework of incremental object learning and tracking for multiple moving objects from RGB-D point set data, as illustrated in Fig. 2. In this framework, each object model is incrementally updated at each time from the identified point data, which are feedback results of the robust tracking process based on the previously constructed object model. This method aims to achieve the three following objectives: flexibility to represent arbitrary objects, robust tracking against interactions between multiple objects, and real-time implementation. In order to achieve flexibility, Gaussian mixture models (GMM) and its tempo-spatial topological graph are used to represent any object shapes and sizes, and their parameters are incrementally updated at each time. The problems of achieving robustness, as illustrated in Fig. 1, is tackled by GMM-based 3-*d* registration for estimating the movements of objects and the multi-frame tracking (MFT) method for constructing robust temporal associations of objects among multiple time frames. The real-time implementation and tracking performance of the proposed method is evaluated and analyzed with several

[*] Corresponding author.
E-mail address: kwonds@kaist.ac.kr (D.-S. Kwon).

1047-3203/$ - see front matter Crown Copyright © 2013 Published by Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jvcir.2013.03.020



(a) *Split*

(b) *Complete occlusion*

(c) *Partial occlusion*

(d) *Multiple contacts*

Fig. 1. Four interaction cases between multiple objects.

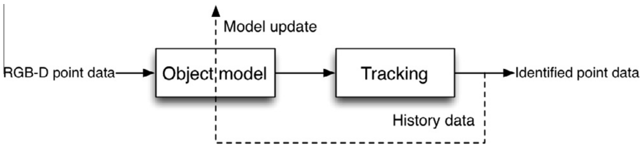


Fig. 2. The framework of incremental object learning and tracking.

experiments involving various movements of human hands with manipulation of multiple objects.

The remainder of this paper is structured as follows. Related work and preliminaries involving object representation and robust tracking methods are described in chapter 2, and an overview of the proposed tracking method and detailed explanations of the processes are provided in chapter 3 and chapter 4, respectively. In chapter 5, the results of the proposed methods are discussed with several experiments. Finally, a conclusion is given in chapter 6.

2. Related work and preliminaries

In this chapter, the preliminaries of the proposed method for incremental object learning and robust multiple object tracking are introduced with a historical research background.

2.1. Object representation from RGB-D point set data

When an object detector is used to identify each object based on pre-learned object models, the tracking problem simply involves collecting identified object information in its track at each time step. In reality, however, the modeling and learning of all objects in advance are not always possible, and new objects that are not modeled can be present while performing a tracking task. In this case, it is necessary to construct a model representing an arbitrary object in online manner.

One approach to solve this problem is to represent an object as a set of primitives. [27,34] constructed each object based on 3-*d* primitive shapes such as a sphere, a plane, a cylinder, and a cone from a point data set. Meanwhile, [32] obtained a more precise object model by combining the primitive shapes and triangular meshes for the remaining point parts. In more recent robotics research, [23] modeled a new object as a set of surfels that is robust to noise and occlusions by using both the shape and appearance information. Another approach is to model the objects as a set of features from the appearances of shape and/or color information. [18] suggested combining a 2-*d* feature-based online boosting tracker and a 3-*d* model-based tracker. In their framework, any object model shape can be constructed in the 2-*d* image domain by using an online multi-class boosting approach, and the 3-*d* position and orientation are estimated by a 3-*d* registration method. They used 6 types of Haar-like features and color features as a feature vector of an object. In order to represent a model of multiple objects, on the other hand, [21] suggested a graphical model to represent the appropriate features of multiple objects, such as supporting contacts, caging, and object geometry for placing the objects in another space.

2.2. Gaussian mixture models

Most approaches in the previous section have been introduced to represent arbitrary rigid objects from the data in several fixed scenes. One of the main objectives of this research is to represent not only rigid objects but also nonrigid or articulated objects in the dynamic situations in the presence of interacting multiple objects. In order to guarantee greater flexibility of the object model in those situations, the model should be flexible and incrementally updatable. We represent an arbitrary object based on a continuous probability density function of a discrete point set involved in the object. This representation is useful for manipulating the object models analytically owing to its functional expression. The most simple case is to design one *d*-dimensional multivariate Gaussian distribution consisting of a mean ($\boldsymbol{\mu} \in \mathbb{R}^d$) and a covariance matrix ($\Sigma \in \mathbb{R}^{d\times d}$) from the *d*-dimensional feature data of an object. The probability density function of the point ($\mathbf{x} \in \mathbb{R}^d$) belonging to the object can be represented as (1).

$$\phi(\mathbf{x}|\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}}\exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right). \tag{1}$$

In particular, the Gaussian mixture model, which is a weighted summation of multiple Gaussians defined as (2), can represent any arbitrary shape of functions when the number of Gaussians (k) goes to infinity.

$$p(\mathbf{x}) = \sum_{i=1}^{k} w_i\phi(\mathbf{x}|\mu_i,\Sigma_i), \quad \sum_{i=1}^{k} w_i = 1. \tag{2}$$

Learning the unknown parameters, $k, \mathbf{w}, \bar{\boldsymbol{\mu}}$, and $\bar{\boldsymbol{\Sigma}}$, from the given data set has been investigated in many ways. One of the typical methods involves using the Expectation-Maximization (EM) algorithm [5,11] given the number of Gaussians (k). In recent years, a hierarchical GMM has been proposed to determine the number of Gaussians efficiently through a hierarchical clustering method [14]. On the other hand, if the assumption that the point set of an object $\mathscr{P} = \{p_1, \ldots, p_n\}$ is obtained using the same sampling distance, the corresponding GMM can be represented by evenly weighted n Gaussians centered at each point with the same spherical covariance matrix [20]. Although a parameter learning process is not needed in such a case, the model includes such massive point data that related algorithms are inefficient due to the expensive computation time.

For this reason, several GMM simplification methods have been suggested. The hierarchical clustering (HC) method [16] utilizes a component grouping algorithm that iterates the process of 'refitting' a Gaussian function to the local group and 'regrouping' points to minimize Kullback–Leibler (KL) divergence between the original GMM and the approximated GMM. Later, this was improved by using a Gaussian-matching clustering algorithm to maximize the cross-entropy approximation between two models [15]. The function approximation (FA) method [41], which used L2 distance measure to minimize the upper bound of the approximation error, showed better performance than [16] in terms of model approximation. [29] suggested a fast algorithm based on the Bregman k-means clustering method to reduce the KL-divergence between two models; they estimated the number of Gaussians, as also done in [14]. In addition, [3] presented a new measurement approach in the multi-class approximation case to maximize the discriminative quality of simplified models among different classes and minimize the similarity between models in the same class.

The similarity measure, which presents the degree to which two GMMs are equivalent, is a useful tool for handling GMMs. It can be a source of cost functions to optimize GMMs in many cases as well as a weight value of an association between two GMMs. Many distance measures between two continuous functions, $g(\mathbf{x}), f(\mathbf{x})$, originates from the density power divergence [4]. The most widely used variants for comparing probability density functions are the KL divergence (3) and the L2 distance (4).

$$d_{KL}(g,f) = \int g(\mathbf{x}) \log \frac{g(\mathbf{x})f(\mathbf{x})}{d}\mathbf{x}, \tag{3}$$

$$d_{L2}(g,f) = \int (g(\mathbf{x}) - f(\mathbf{x}))^2 d\mathbf{x}. \tag{4}$$

These two measures have a trade-off relation between robustness and asymptotic efficiency [20]. In terms of comparing two probabilistic density functions, the robustness means that the distance is not likely affected by the global change of the functions, but it can reflect the local properties of the functions. The asymptotic efficiency has the opposite meaning. These properties of two measures show different behaviors in the distance of two functions with respect to the change of one function: the L2 distance has an exact global minimum with many local minima, but the KL distance has one minimum point that is biased by global effects. Therefore, the KL distance is appropriate for estimating the distance in the cases of two functions with a large difference, and the L2 distance can reflect more exact distance in the case of relatively close functions around the global minimum.[1]

2.3. Robust tracking methods for a single target

The conventional tracking method of a single object uses a bayesian filtering method to stochastically estimate the current position of an object from the past observations with transition and measurement noise models. Many kinds of filtering methods have been developed according to the modeling of the target object [2]. Once it is possible to use the useful information of the dynamics of moving objects, the estimation can be achieved by Kalman-based filtering methods while particle filtering-based methods can be applied in the cases of tracking objects without dynamic models. In [25,42], particle filtering was performed to track a human body with a pre-defined skeleton model while any kind of object can be tracked using particle filtering with an on-line constructed model [39]. However, particle filtering can be applied to observed data only with outliers corresponding to the noise model. When the observed point data is obtained from multiple objects with arbitrary outliers, particle filtering should be applied after the observed data is segmented as a source for each target.

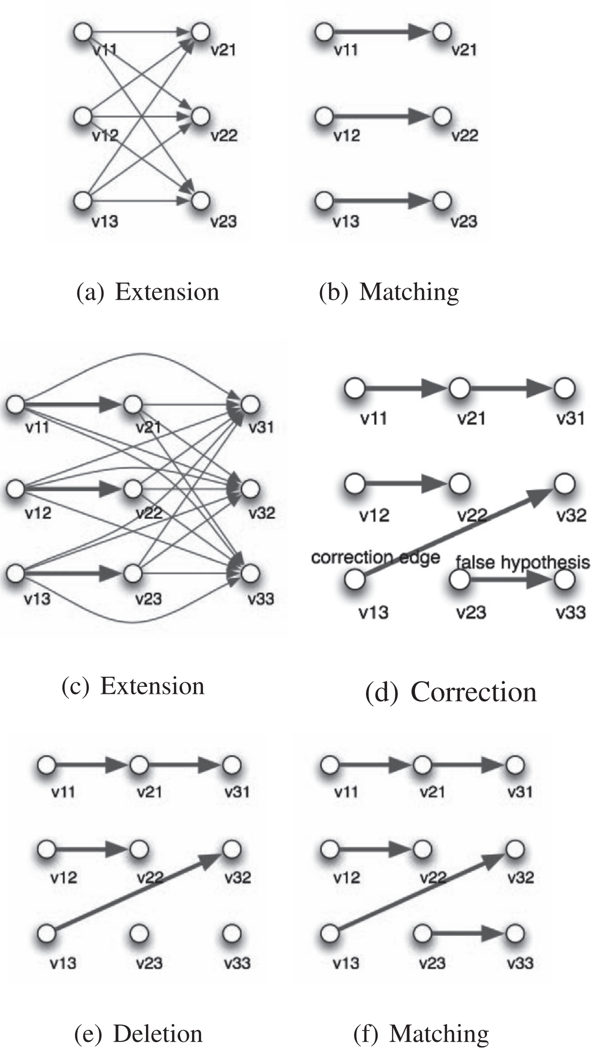
In particular, object tracking from 3-*d* point set data with many outliers can be achieved using 3-*d* registration methods which match the shapes of two point data sets by transforming a set of points to another in 3-*d* space. This is data-based optimization method that employs the assumption that there is one global minimum point of 3-*d* transformation parameters. The most well-known method is the Iterative Closest Point (ICP) algorithm, which has many variants [31]. In the case of tracking non-rigid objects such as the human body, an articulated ICP with a structure model was introduced in [10,22,23]. The structure model of the target object facilitates robust tracking in the presence of occlusions or outliers. In the case of tracking multiple arbitrary objects, where the structure models of objects are not available, the measurement of the target object should be refined from all detected points by rejecting occluded points or outliers. [18] used the ray-casting approach, where 2-*d* projections of the reference points to the observed points are used to detect the occlusions. On the other hand, for robust tracking with outliers, a GMM-based robust registration method was proposed in [19] without an outlier rejection process. [20] showed the robustness of the method compared with the conventional ICP method in a situation of considerable outliers and occlusions by introducing the robustness of the L2-distance.

2.4. Multi-frame tracking (MFT) for robust tracking multiple targets

In the case of tracking multiple objects, the objects at the previous time frames should be associated with those at the current time frame. Temporal data associations between time frames encompass the issues of a variable number of tracks, the initialization and termination of tracks, and false matching. Joint probabilistic data association filter (JPDAF) and multi-hypothesis tracking (MHT) methods are probabilistic approaches for the temporal matching of objects according to frames. They calculate the probability of each track for all possible matches with a probability density function around new points. Although JPDAF assumes a fixed number of tracks [35], MHT is extended to work with a variable number of tracks, and it has been used in many applications [26,24]. Although the probabilistic approach has been applied successfully in many tracking applications, it is associated with a number of intractable problems such as the assumption of a probability density function for all points, sensitivity of the tracking performance to the number of parameters of the model, and the computational complexity growing exponentially with the number of points. On the other hand, several deterministic approaches have been proposed to overcome the probabilistic approaches. The greedy optimal assignment (GOA) algorithm [40] was shown to enhance the performance of finding optimal associations to allow occlusions and for detection errors with a constant number of points. Shafique and Shah [36] proposed the multi-frame tracking (MFT) algorithm to improve the tracking performance by considering point information in multiple frames for a variable number of points.

MFT is an efficient and robust temporal data association method based on the noniterative greedy algorithm [40]. The maximum matching algorithm among multi-frame data allows correction of existing correspondences, which compensates occlusions and detection errors of targets. Fig. 3 summarizes the processes of the MFT method in [36]. In the first two time frames, there are three objects in the first frame (v_{11}, v_{12}, v_{13}) and in the second frame (v_{21}, v_{22}, v_{23}), respectively. Extension edges can then be connected to all objects between the two frames, as shown in Fig. 3(a). Based on the weight values at each edge, the maximum matching algorithm is performed to find optimal correspondences, as indicated by the bold arrows in Fig. 3(b). When objects in a new frame enter the graph, extension edges are generated from all objects in the existing *k*-frames, as shown in Fig. 3(c). These extended edges can share objects with existing correspondences, resulting in correction edges and false hypotheses after maximum matching. Fig. 3(d) shows the correction edge $(v_{13} - v_{32})$ of the previous correspondence $(v_{13} - v_{23})$ and the false hypothesis $(v_{23} - v_{33})$ caused by the false correspondence $(v_{13} - v_{23})$. Because the false hypothesis is meaningless with the correction edge, it is removed in the correspondences, as shown in Fig. 3(e). After the deletion step, the remaining unconnected objects perform maximum matching between adjacent frames, as presented in Fig. 3(f). These processes allow corrections of existing correspondences between

[1] In this research, L2 distance is used for evaluating the temporal associations between two GMMs, and KL divergence is used for evaluating spatial associations between two Gaussians. This is explained in chapter 4 in detail.



(a) Extension (b) Matching

(c) Extension (d) Correction

(e) Deletion (f) Matching

Fig. 3. Processes of multi-frame tracking algorithm.

objects which compensates for occlusion and detection errors of objects.

The size of the multi-frames, k, serves as a sliding window in which all object histories of all tracks are extended to the objects in a new frame. This means that corrections of existing correspondences are possible in the window. If k is larger than the occlusion time, tracks that have disappeared or that are mismatched can be recovered as newly detected objects after the occlusion process.

3. Problem statements and the framework of the proposed method

TThe main task of tracking multiple objects from RGB-D point set data is to identify each instance of point data at each time, $\mathscr{P}^t = \{p_1^t, \dots, p_n^t\}$, to each true object track, $\mathscr{T}^t = \{\{p_1^t, t_1^t\}, \dots, \{p_n^t, t_n^t\}\}$, in situations with the following three areas of difficulty. The first problem is the changeability of the object shape. In dynamic situations, a non-rigid object such as a human hand can change in terms of its shape and size, and adjoining multiple objects cause distortion of each type of detected object shape, as shown in the *multiple contacts* case in Fig. 1(d). The second difficulty is the separation of an object into several objects, as demonstraed in the cases of *split* and *partial occlusion* shown in Fig. 1(a) and (c). In both cases, the object is divided into two parts, and the two parts should be identified as individual objects in the *split* case. In contrast, they must be recognized as one object in the *partial occlusion* case. The third problem is the change in the number of objects. In the cases of the *complete occlusion* of a small object covered by a larger object and objects coming in and out of the scene, the number of detected objects varies.

In order to tackle the three problems as stated above, the proposed robust multiple object tracking algorithm consists of three steps: measurement estimation, incremental object learning, and multiple object tracking, as shown in Fig. 4. Measurement estimation is the process of identifying a set of newly observed RGB-D points that is associated with each object. At each time step, the Maximum weighted Likelihood (MwL) of each instance of point data is evaluated with predictive object models that are estimated from previous object models using GMM-based 3-*d* registration (GMM-Reg). An identified point data group within an object updates the corresponding object model incrementally. Each object is represented by a GMM[2] from its point data using a GMM simplification method. Each Gaussian in the updated GMM is temporally associated with the previous GMM in an identical predictive object using the MFT algorithm at the Gaussian-level (MFT-G) in order to construct a temporal-spatial topological graph (TSTG) of the object.

Multiple object tracking is the process of constructing robust temporal associations between the updated object models. In cases where there exist new objects in the scene such as *split*, as in Fig. 1(a), and objects newly entering the scene, the updated object models are investigated and separated into individual new objects. In order to assign a track id t_i^t to each object O_i^t at each time step, the multi-frame tracking algorithm at the GMMs-level (MFT-GMM) determines the multi-object temporal association using a similarity measure between the GMMs of objects, thus resulting in $\mathscr{T}^t$. This process can handle typical tracking problems of generating new tracks, deleting old tracks, and correcting false matches due to noise and occlusions.

The main contributions of the proposed method for robust tracking of multiple objects without prior knowledge are summarized as follows:

- The proposal of a flexible object model based on a GMM with a tempo-spatial topological graph (TSTG) and its incremental learning method to represent arbitrary objects.
- The robust identification of all observed point data to be involved in true individual objects by using robust 3D registration (GMM-Reg) and Maximum weighted Likelihood (MwL) point matching.
- The development of robust temporal associations of multiple objects in multiple frames in the cases of various interactions between them.

In the following chapters, the details of the components in the proposed framework are described and their performance and limitations are evaluated with various experimental settings.

4. The proposed multiple object tracking method

4.1. Measurement estimation

In order to identify each point in newly observed point set data, $\mathscr{P}^t = \{p_1^t, \dots, p_n^t\}$, as its object to be involved, $\mathscr{O}^t = \{\{p_1^t, o_1^t\}, \dots, \{p_n^t, o_n^t\}\}$, it is necessary to estimate the predictive object at time t that is represented by a GMM[3], $\widehat{gmm}_i^t$, from the existing object

[2] Any dimension of point data can be applicable. In this research, 3-dimensional position (x, y and z) data and 6-dimensional position and color (r, g, and b) data of a point are applied and their performance is investigated in chapter 5.

[3] The construction of a GMM, gmm_i from the point data set O_i is described in chapter 4.2.

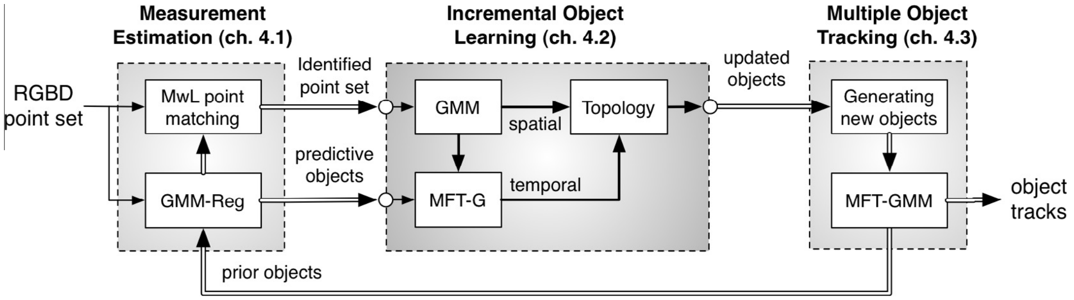
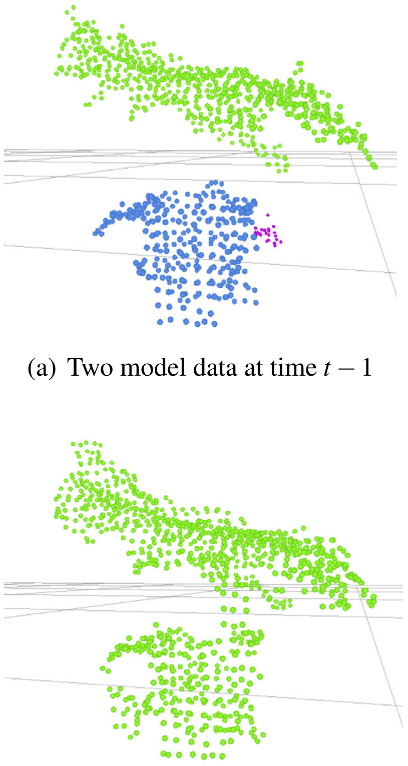


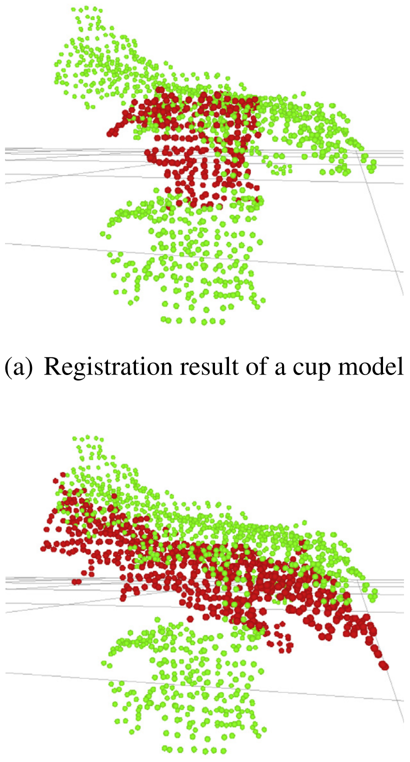
Fig. 4. Overview of the proposed tracking processes.



(a) Two model data at time $t-1$

(b) A scene data at time t

Fig. 5. 3D point set registration task.



(a) Registration result of a cup model

(b) Registration result of a hand model

Fig. 6. 3D point set registration results using KL-divergence of the two models.

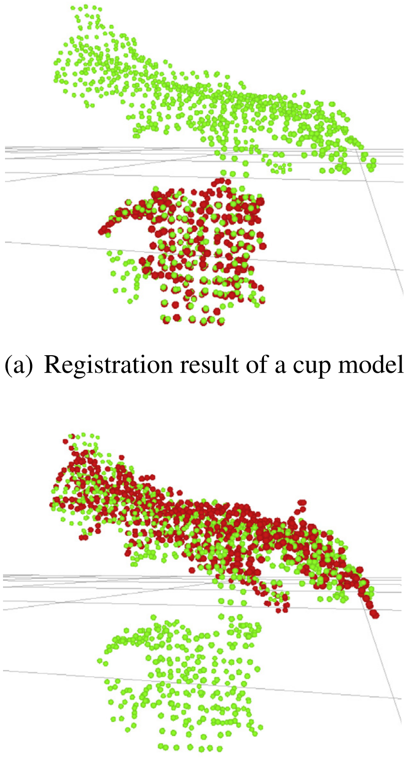
model at $t-1$, gmm_i^{t-1}, needs to be transformed to estimate the predictive object at time t, $\widehat{gmm}_i^t$. Because there is no prior information of the object movements, in this research, the prediction process can be achieved by using the robust 3-*d* point set registration method with the assumption that there is no substantial shape change of an object between concatenated time frames.[4]

The 3-*d* point set registration is defined as a process that finds the transformation parameter θ to minimize the distance or maximize the similarity between the point set of the transformed model, $T(P_m, \theta)$ and the point set of the scene P_s. In this study, we use GMM from the point set as an object to reduce the data size and thereby enhance the computational efficiency. GMM can be applied for ICP as a kernel function, as in [8,38]; [19] showed that ICP-based registration methods have the same effect of minimizing the KL-divergence between two GMMs of $T(P_m, \theta)$ and P_s. The KL-divergence of two GMMs of (3) is an efficient metric for comparing two point sets, but is not robust in the presence of outliers. [20] proposed a GMM-based registration method using the L2 distance of GMMs as a cost function between the transformed model and the scene. The L2 estimator is more robust against outliers than the KL-divergence estimator and the maximum likelihood estimator (MLE). Another advantage of the L2 distance is its closed-form expression for GMMs. The L2 distance of two GMMs can be expressed as (5) with the property of a Gaussian function of (1), $\int \phi(\mathbf{x}|\mu_1, \Sigma_1)\phi(\mathbf{x}|\mu_2, \Sigma_2)d\mathbf{x} = \phi(\mathbf{x}|\mu_1 - \mu_2, \Sigma_1 + \Sigma_2)$.

$$\begin{aligned} d_{L2}(g,f) &= \int f^2(\mathbf{x})d\mathbf{x} - 2\int f(\mathbf{x})g(\mathbf{x})d\mathbf{x} + \int g^2(\mathbf{x})d\mathbf{x} \\ &= \sum_{i=1}^{m}\sum_{j=1}^{m} w_i^f w_j^f \phi(0|\mu_i^f - \mu_j^f, \Sigma_i^f + \Sigma_j^f) \\ &\quad - 2\sum_{i=1}^{m}\sum_{j=1}^{n} w_i^f w_j^g \phi(0|\mu_i^f - \mu_j^g, \Sigma_i^f + \Sigma_j^g) \\ &\quad + \sum_{i=1}^{n}\sum_{j=1}^{n} w_i^g w_j^g \phi(0|\mu_i^g - \mu_j^g, \Sigma_i^g + \Sigma_j^g). \end{aligned} \tag{5}$$

[4] The breaking conditions of this assumption are empirically tested in chapter 5.2.

Please cite this article in press as: S. Koo et al., Incremental object learning and robust tracking of multiple objects from RGB-D point set data, J. Vis. Commun. (2013), http://dx.doi.org/10.1016/j.jvcir.2013.03.020



(a) Registration result of a cup model

(b) Registration result of a hand model

Fig. 7. 3D point set registration results using L2-distance of the two models.

The numerical calculation of (5) consists of three forms of discrete Gaussian transforms [17], and the performance in terms of the computation time depends mainly on the number of Gaussians, *m* and *n*. Hence, a reduction of the size is necessary for implementation of the algorithm in real-time.[5]

The goal of the registration is to transform the past true objects (models) into an observed point set (scene) to minimize the differences between their GMMs. As shown in Fig. 5(a), the two objects (a human hand and a cup) have previous object models at time $t-1$ but are contacted at time *t*, as shown in Fig. 5(b). The two point sets of objects in Fig. 5(a) are the model data, and the observed point set in Fig. 5(b) is the scene data for 3D registrations to estimate the true object point set of each object at time *t*.

For each model, the scene data have numerous outliers, which are points belonging to another object; therefore, a robust registration method is preferable in this case. Fig. 6 shows the GMM-based registration results with the KL-divergence distance while Fig. 7 shows the results when the L2-distance. Obviously, the KL-divergence measure more efficiently reflects the global effects of the points, as it tries to maximize the likelihood of the model matching the scene and thus places the model at the center of the scene. On the other hand, the L2-distance reflects local effects better than it shows a global influence, and the registration results show that it is more robust against most outliers.

Another advantage of GMM-based 3D registration[6] is its closed expression of the gradient of the cost function. In this research, we used rigid transformation, which is defined by the rotation matrix $\mathbf{R}$ and the translation vector $\mathbf{t}$. Let $\mathbf{P}$ denote a $n \times d$ matrix of the d-dimensional point set $\mathscr{P}$. The rigidly transformed model at time *t* can then be expressed as follows:

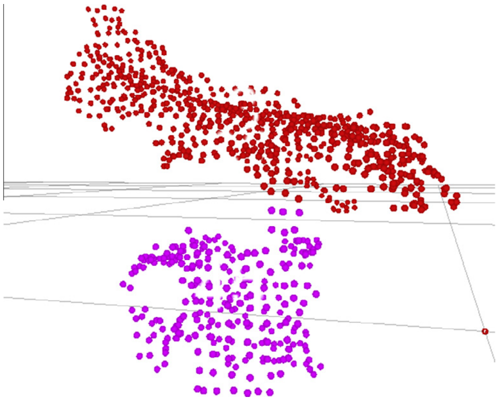


Fig. 8. The result of correcting the false segments in the Fig. 5(b).

$$\mathbf{P}^t_m = T(\mathbf{P}^{t-1}_m, \theta) = \mathbf{P}^{t-1}_m \mathbf{R}^T + \mathbf{t}. \tag{6}$$

The gradient of the cost function (5) can be derived by the chain rule $\frac{\partial F}{\partial \theta} = \frac{\partial F}{\partial \mathbf{P}^t_m}\frac{\partial \mathbf{P}^t_m}{\partial \theta}$. The first derivative $\frac{\partial F}{\partial \mathbf{P}^t_m}$ is the partial derivative of the cost function with respect to each point. The derivatives of the first and the third terms of (5) are zero due to the rigid transformation. The partial derivative of the cost function at each point is determined as follows:

$$\frac{\partial F}{\partial \mu^m_{i,d}} = -2w^m_i \sum_{j=1}^{n} w^s_j \frac{\partial}{\partial \mu^m_{i,d}} \phi(0|\mu^m_i - \mu^s_j, \Sigma^m_i + \Sigma^s_j). \tag{7}$$

The second derivative can be simply obtained by the linear form of (6). The gradient of the cost function can be expressed as

$$\begin{aligned} \frac{\partial F}{\partial \mathbf{t}} &= \frac{\partial F}{\partial \mathbf{P}^t_m}^T \mathbf{1}_m, \\ \frac{\partial F}{\partial r_i} &= \mathbf{1}^T_d \left(\left(\frac{\partial F}{\partial \mathbf{P}^t_m}^T \mathbf{P}^{t-1}_m \right) \otimes \left(\frac{\partial \mathbf{R}}{\partial r_i} \right) \right) \mathbf{1}_d, \end{aligned} \tag{8}$$

where $\mathbf{1}_m$ is a *m* dimensional column vector of all ones, and $\otimes$ denotes element-wise multiplication. In order to optimize the transformation parameter, any gradient descent optimization algorithm can be used with the help of the gradient of (8). In this research, we used the limited-memory Broyden Fletcher Goldfarb Shannon (L-BFGS) minimization algorithm, which is based on a quasi-Newton algorithm for large-scale numerical optimization problems.[7]

After the registration process, each predictive object at time *t*, $\widehat{gmm}^t_i$, can be estimated, as shown in Fig. 7(a) and (b). All of the observed points can then be evaluated in terms of the degree to which each is involved with $\widehat{gmm}^t_i$ by comparing the likelihoods of the GMMs at each point. Here, o^t_i is the identification number of an object to which point p^t_i is related; it can be determined by the Maximum Likelihood Estimator (MLE) of GMMs as

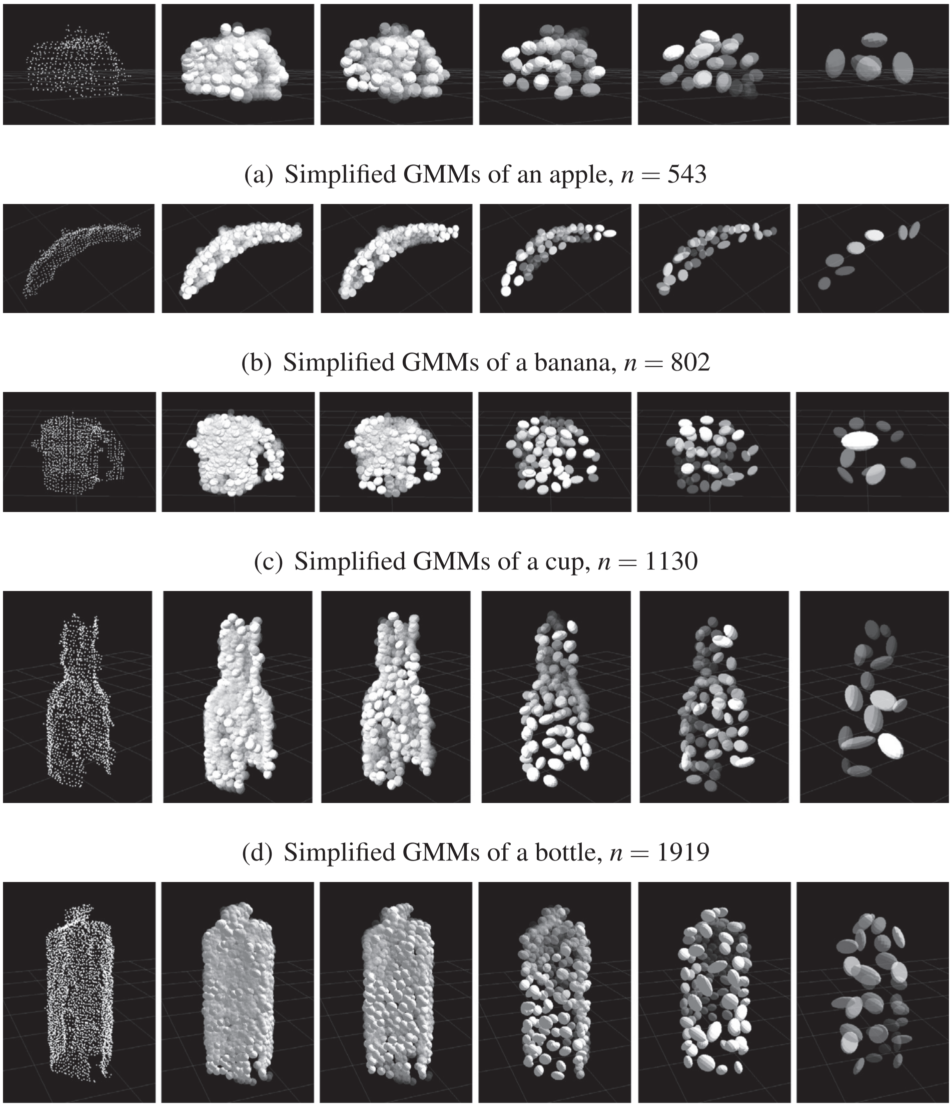
$$o^t_i = \arg\max_j L(\widehat{gmm}^t_j | p^t_i) = \arg\max_j \widehat{gmm}^t_j(p^t_i), \tag{9}$$

where $gmm(p)$ is the evaluated value of a GMM at point *p* according to (2). Although the MLE can evaluate and compare multiple GMMs at one point, it has an inherent problem when used to compare GMMs with different numbers of components. Due to the property

[5] The computation time according to the number of Gaussians is discussed in chapter 5.2.

[6] The details of the GMM-based 3-*d* registration method are explained in the literature [20]. The algorithm is briefly summarized in terms of its advantages in this research.

[7] This is implemented in the vision-numerics library (vnl) in <http://vxl.sourceforge.net/>.



(a) Simplified GMMs of an apple, $n = 543$

(b) Simplified GMMs of a banana, $n = 802$

(c) Simplified GMMs of a cup, $n = 1130$

(d) Simplified GMMs of a bottle, $n = 1919$

(e) Simplified GMMs of a juice pack, $n = 3361$

Fig. 9. The first column denotes the n source points. The simplified GMMs are displayed as a set of 3-d ellipsoids with reduction ratios of 0.5, 0.3, 0.1, 0.05 and 0.01, respectively, from the second to the sixth column.

of the weights in GMM, $\sum_{i=1}^{k} w_i = 1$, GMM with more Gaussians has a lower evaluation value at the given point. This property results in biased evaluations of the points at the boundaries of two GMMs, which are more likely associated with GMM with smaller Gaussians. For this reason, we propose the Maximum weighted Likelihood (MwL) function of GMM by normalizing the number of components as shown below,

$$o_i^t = \arg\max_j L(\widehat{gmm}_j^t|p_i^t) = \arg\max_j \frac{n_j^t \times \widehat{gmm}_j^t(p_i^t)}{\sum_j n_j^t}, \tag{10}$$

where n_j^t is the number of Gaussians in gmm_i^t. Fig. 8 shows the result after MwL point-matching.

4.2. Incremental object learning

4.2.1. Gaussian mixture models construction

The set of identified points that carries the same value of $o_i^t = k$ is clustered as one object O_k^t, where n_o^t is the number of objects at time t and $n_{o,k}^t$ is the number of points identified as the object k at time t.

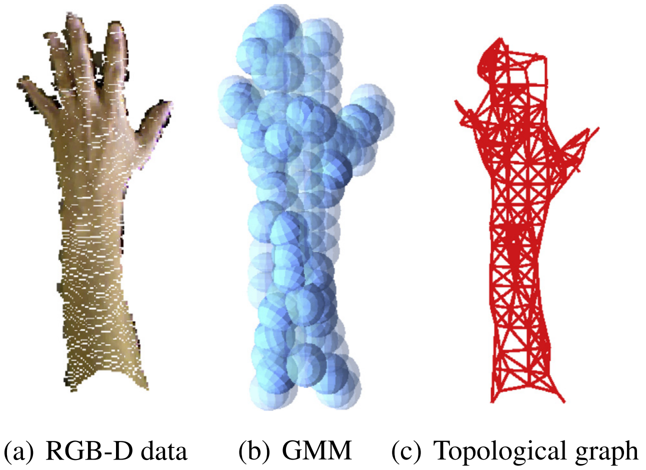
$$\mathcal{O}^t = \{O_1^t, \ldots, O_{n_o^t}^t\}, \quad O_k^t = \left\{\{p_1^t, k\}, \ldots, \{p_{n_{o,k}^t}^t, k\}\right\}. \tag{11}$$

Each point set of object O_k^t constructs a GMM that consists of the same number of Gaussians as the number of points with a constant diagonal covariance matrix determined by the sampling distance, σ.[8]

$$p(\mathbf{x}) = \sum_{i=1}^{n_{o,k}^t} w_i \phi(\mathbf{x}|\mu_i, \Sigma_i), w_i = \frac{1}{n_{o,k}^t}, \quad \mu_i = \mathbf{x}_i, \Sigma_i = diag(\sigma) \in \mathbb{R}^{d\times d}. \tag{12}$$

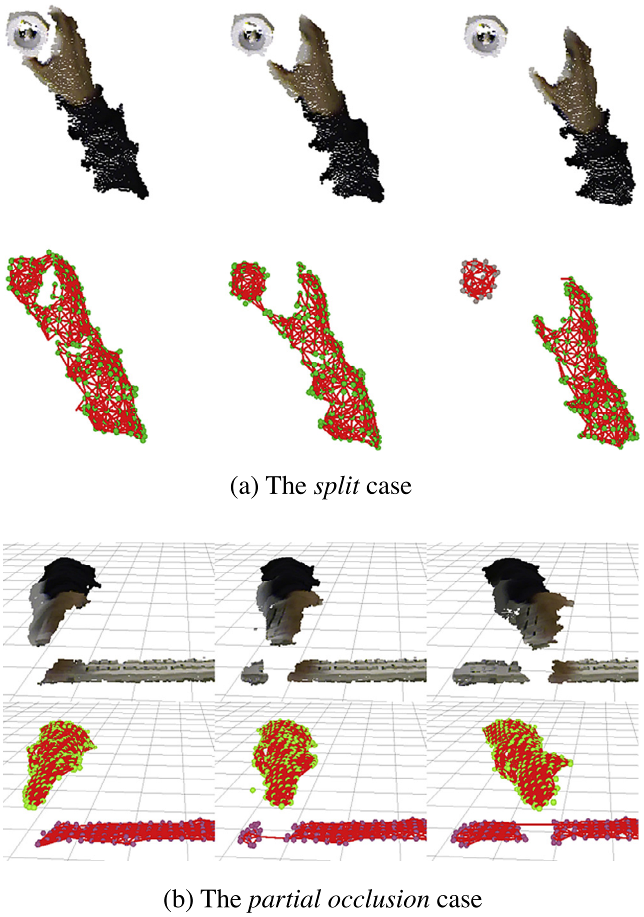
The initially constructed GMM with a size of $n_{o,k}^t$ is approximated by means of a simplification method. The method approximates a GMM with a given number of Gaussians that is proportional to the number of points involved with the object, which determines

[8] The sampling distance is a pre-defined value for capturing point data from the camera. In chapter 5.1, the change of this value and its relation with the tracking performance is presented.



(a) RGB-D data (b) GMM (c) Topological graph

Fig. 10. The constructed GMM and topology from RGB-D data of a human hand.



(a) The *split* case

(b) The *partial occlusion* case

Fig. 11. The sequential change (from left to right) of the topological graph in the *split and partial occlusion* cases. The upper rows show the captured RGBD point set data, and the lower rows illustrate GMMs with their topological graphs.

the trade-off between the computation time and the model approximation error.[9]

Fig. 9 shows the simplified GMMs of five objects in different point sizes. The point set of the objects (an apple, a banana, a cup, a bottle, and a juice pack) was captured by two Kinect cameras and were down-sampled with a distance of 0.005 m between each point, as depicted in the first column of Fig. 9(a)–(e). The point set generates an initial GMM with the same number of points n and a diagonal covariance matrix with $\sigma = 0.005$. The initial GMM of an object was reduced to a mixture of m Gaussians ($m < n$), according to the HC method[10] with different reduction ratios m/n ranging from 0.01 to 0.5. The second to the sixth columns in Fig. 9 express the GMM simplified by a set of 3-*d* ellipsoids with different transparent values according to the weighted value of the Gaussian. Each ellipsoid locates at the centroid of each Gaussian, and the 3-*d* size and the orientations are calculated from the eigenvalues and the corresponding eigenvectors of the covariance matrix of the Gaussian.

4.2.2. Multi-frame tracking in Gaussians (MFT-G)

In order to increase the robustness of the model, motion information of individual object is a very strong cue in separating objects [13,30]. For example, in the *split* and *partial occlusion* cases in Fig. 1(a) and (c), one object can be recognized as two separated objects. The movements of the separated components in an object can be used to distinguish these two cases by comparing their velocities. The velocity of each Gaussian can be generated in the temporal associations by tracking the history of the position of the Gaussian. Therefore, the Gaussians in the GMM must have not only spatial but also temporal relations among them.

The temporal associations can be developed by the MFT algorithm presented in chapter 2. In an object at each time frame t, each Gaussian is a new node in the t frame of the graph of MFT as nodes v_{21}, v_{22}, and v_{23} in Fig. 3(a). The only necessary parts to construct a MFT for Gaussians are the definition of the weight function between two Gaussians and the size of the time frame k to extend the new associations. k is determined by the given situations to consider the length of partial occlusion time. The weight function between two Gaussians, $g_1 = \{\mu_1, \Sigma_1\}$ and $g_2 = \{\mu_2, \Sigma_2\}$, is determined by using the L2 distance between them as follows.

$$weight(g_1, g_2) = 1 - \frac{d_{L2}(g_1, g_2)}{\max\limits_{i,j}(d_{L2}(g_i, g_j))}. \qquad (13)$$

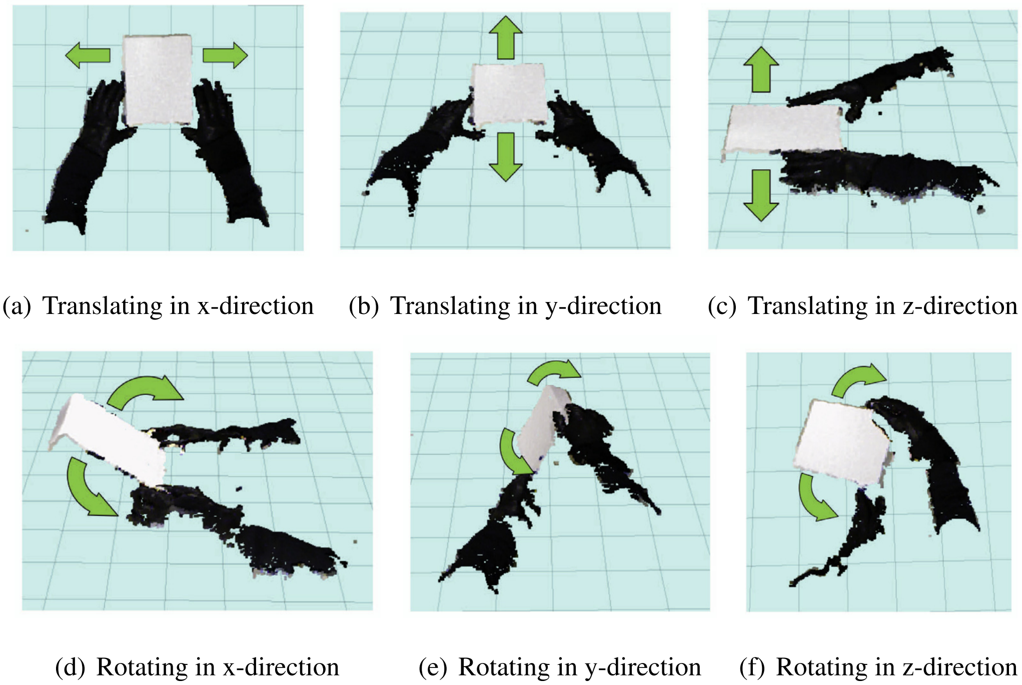
Because the matching algorithm in MFT maximizes the sum of weight values in the matched associations, the L2 distance, that is 0 for the closeness Gaussians, is converted to a weight value between 0 and 1 by introducing the maximum value of the distance in the graph.

4.2.3. Tempo-spatial topological graph (TSTG) construction

In order to represent the spatial and temporal relations among the Gaussians in a GMM, the GMM constructs a topological graph where each node represents each Gaussian and the undirected edge between two nodes. In formal expression, the topological graph can be expressed as $G = \{V, E\}$. The graph contains as many nodes as the number of Gaussians in the object, $v_i \in V,\ 1 \leqslant i \leqslant m$. Each undirected edge of the graph, $e_{i,j} \in E,\ 1 \leqslant i < m,\ i < j \leqslant m$, represents the association between two Gaussians of v_i and v_j. Each edge contains the weight value of the association. In order to reflect the spatial and temporal relations between two Gaussians, the weight value of an edge is determined as a convex combination of the differences of posi-

[9] The ratio between the number of Gaussians and the number of points is tested and the performances of the two methods are investigated and compared through various experiments in chapter 5.1.

[10] Considering two well-known simplification methods, FA [41] and HC [16], HC is more appropriate than FA in this research due to the difference in their computation times with similar approximation accuracy levels.



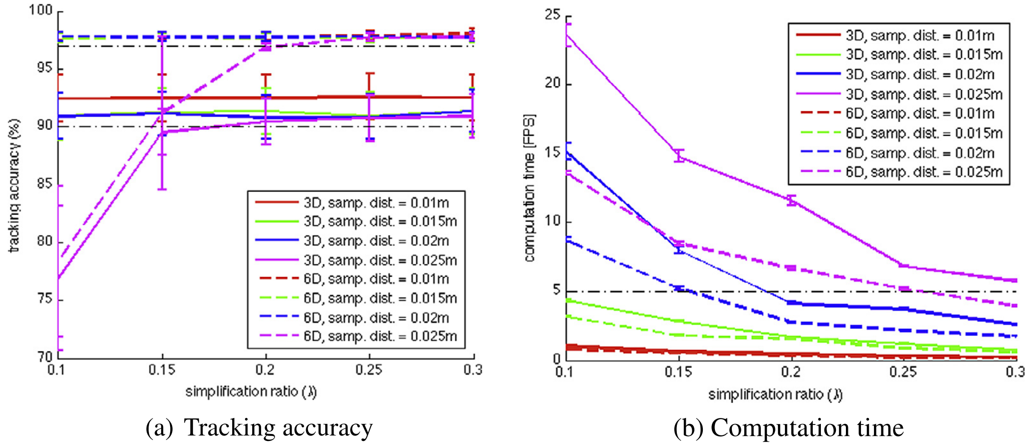
(a) Translating in x-direction (b) Translating in y-direction (c) Translating in z-direction

(d) Rotating in x-direction (e) Rotating in y-direction (f) Rotating in z-direction

Fig. 12. Six hand motions with a white box in the presence of multiple contacts.

Table 1
Average number of points at each frame of the six test data according to the sampling distance (from 0.01 m to 0.025 m).

Task	Time [s]	# Of frames	Average # of points with a sampling distance [m] of			
			0.01	0.015	0.02	0.025
Translation in *x*	24.9027	794	1518.9567	713.8847	424.1239	293.1153
Translation in *y*	20.8707	590	1377.9966	655.1016	381.8983	274.2372
Translation in *z*	24.7281	698	1158.3763	574.6112	350.6074	245.5587
Rotation in *x*	26.4981	756	1525.8542	721.9603	429.5926	300.1667
Rotation in *y*	20.8379	568	1468.0352	704.3133	420.1901	296.3767
Rotation in *z*	21.2505	600	1260.2286	603.6357	359.9178	256.2357



(a) Tracking accuracy (b) Computation time

Fig. 13. Averaged tracking accuracy and computation time results of the six hand motions translating and rotating a white box in the presence of multiple contacts.

tion and velocity between two Gaussians, which is controlled by a parameter $0 \leqslant \alpha \leqslant 1$.

$$w(e_{i,j}) = \alpha \times w_{pos}(e_{i,j}) + (1 - \alpha) \times w_{vel}(e_{i,j}). \tag{14}$$

The position difference of two Gaussians is defined by the normalized KL distance in an object and converted into a weight of association between 0 and 1 as follows.

$$w_{pos}(e_{i,j}) = 1 - \frac{d_{KL}(g_i, g_j)}{\max_{i,j}(d_{KL}(g_i, g_j))}. \tag{15}$$

In the case of defining the relation between spatially distributed Gaussians, the KL distance is more appropriate than the L2 distance due to its global effect. The symmetrised KL distance (16) is used for the undirected edges.

Please cite this article in press as: S. Koo et al., Incremental object learning and robust tracking of multiple objects from RGB-D point set data, J. Vis. Commun. (2013), http://dx.doi.org/10.1016/j.jvcir.2013.03.020

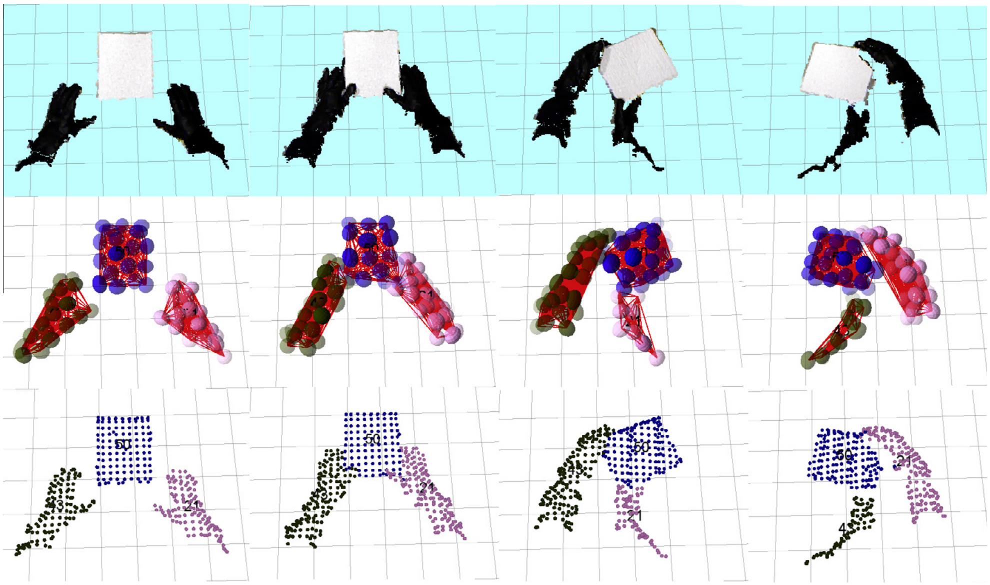
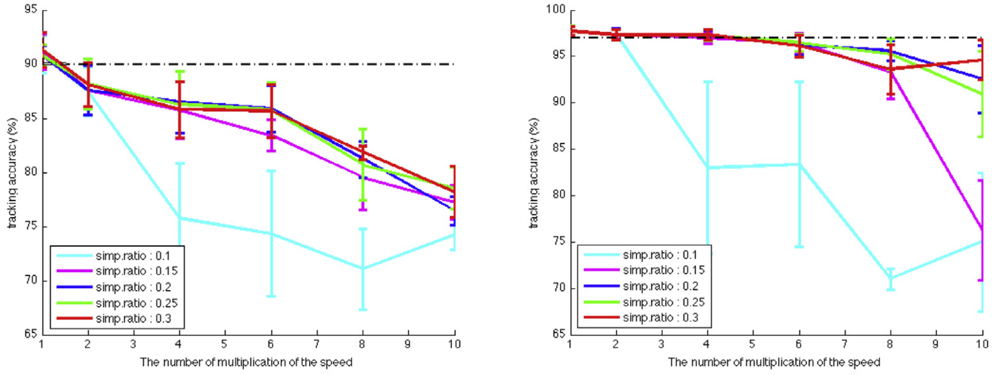


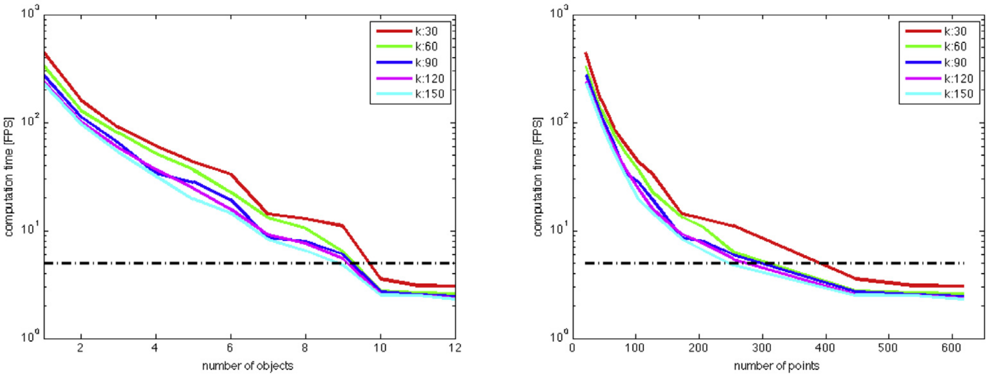
Fig. 14. Illustrations of the tracking results in the sequence (from left to right) of the movement rotating in the z-direction. The first row show the original captured point set data. The second row illustrate Gaussian mixture models as a set of 3D ellipsoids with tempo-spatial topological graph. The tracking results of the proposed algorithm are depicted in the figures on the third row.



(a) 3-*d* GMM

(b) 6-*d* GMM

Fig. 15. Tracking accuracy results of 3-*d* GMM and 6-*d* GMM according to the increasing speed of objects.

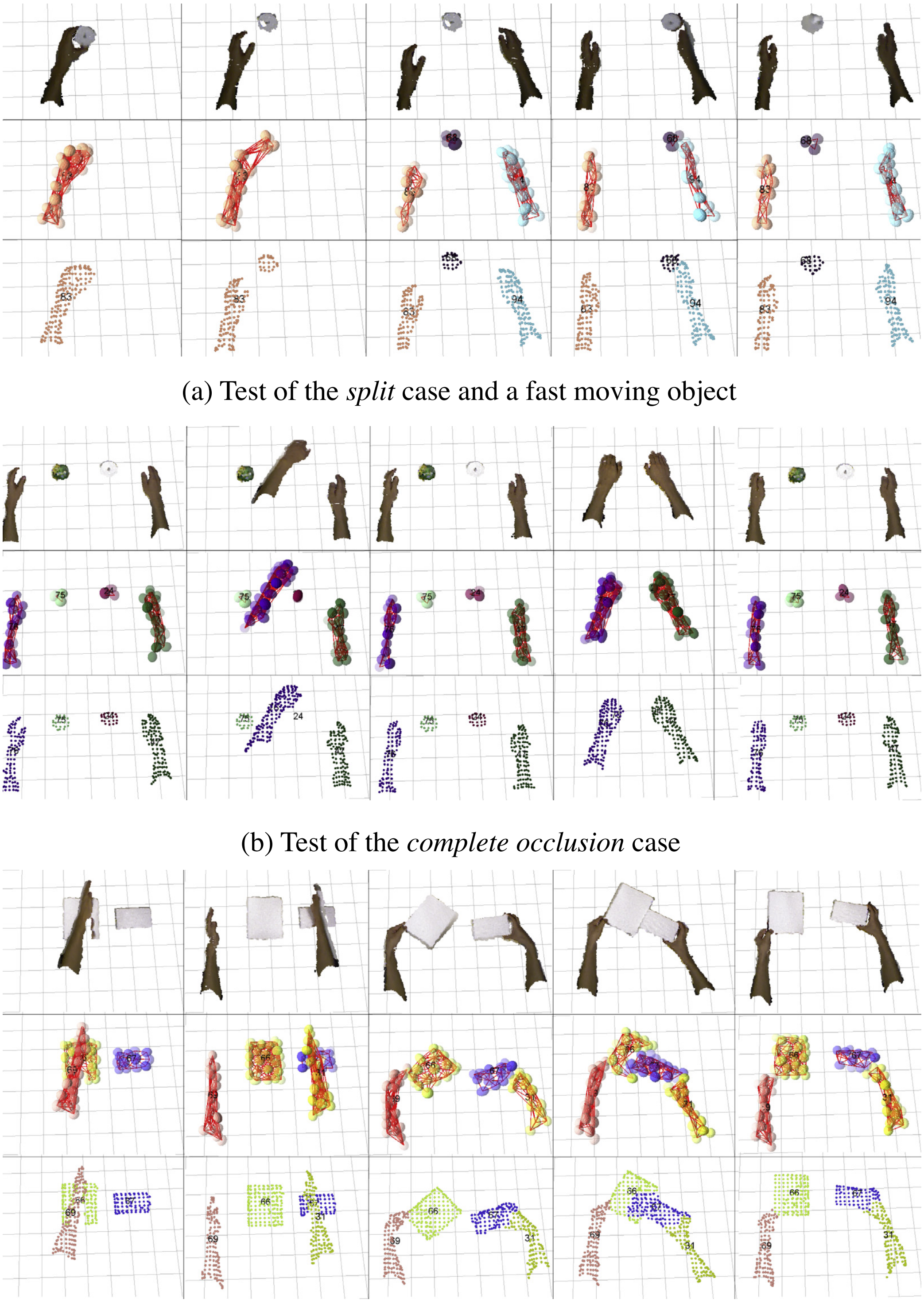


(a) Computation time according to the number of objects

(b) Computation time according to the number of points

Fig. 16. The computation time according to the number of objects and points and the length of frames, *k* in MFT-GMM.

Please cite this article in press as: S. Koo et al., Incremental object learning and robust tracking of multiple objects from RGB-D point set data, J. Vis. Commun. (2013), http://dx.doi.org/10.1016/j.jvcir.2013.03.020



(a) Test of the *split* case and a fast moving object

(b) Test of the *complete occlusion* case

(c) test of the *partial occlusion* and *multiple contacts* cases

Fig. 17. Snapshots of the tracking multiple objects in the sequence (from left to right) of the movements. The first row of each figure show the original captured point set data. The second row illustrate Gaussian mixture models as a set of 3D ellipsoids with tempo-spatial topological graph. The tracking results of the proposed algorithm are depicted in the figures on the third row.

$$d_{KL}(g_1, g_2) = d_{KL}(g_1 \| g_2) + d_{KL}(g_2 \| g_1), \quad \text{where}$$
$$d_{KL}(g_1 \| g_2) = \frac{1}{2}\left(tr(\Sigma_2^{-1}\Sigma_1) + (\mu_2 - \mu_1)^T \Sigma_2^{-1}(\mu_2 - \mu_1) - \ln\left(\frac{|\Sigma_1|}{|\Sigma_2|}\right) - d\right). \quad (16)$$

The weight value for referring temporal properties of Gaussians is a normalized velocity difference between two Gaussians. Because each Gaussian already has a historical track from the MFT algorithm, the velocity can be calculated by the change of position vectors in a Gaussian as follows.

$$w_{vel}(e_{i,j}) = 1 - \frac{d_{vel}(g_i, g_j)}{\max\limits_{i,j}(d_{vel}(g_i, g_j))}, \quad \text{where}$$
$$d_{vel}(g_i, g_j) = \left|(\mu_i^t - \mu_i^{t-1}) - (\mu_j^t - \mu_j^{t-1})\right|. \tag{17}$$

The fully connected topological graph with initial weight values of all edges needs to be simplified to construct a meaningful topology of the GMM. The weight value of each edge is tested with a threshold value, $0 < th_{edge} < 1$, to erase the edge in the graph.

$$w(e_{i,j}) < th_{edge} \Rightarrow erase\ e_{i,j} \text{ in } E. \tag{18}$$

Fig. 10 shows an example of the constructed GMM and its topology graph of a human hand.

4.3. Multiple object tracking

There are two types of new objects: a separated part in the *split* case and an object newly entering into the scene (*entering* case). In order to generate new objects in these cases, topological graphs in the existing objects are investigated. Because the weight value of an edge represents the closeness of two Gaussians in terms of the spatial positions and temporal movements, the two individual objects are easily disconnected when they move in different ways, as in the *split* case, or they are positioned at a distance from each other in the *entering* case. On the other hand, even if there is partial occlusion of an object, the Gaussians are not easily disconnected when they have the same movement patterns. Therefore, the disconnected parts[11] in the topological graph are generated as new objects in the *split* and *entering* cases.

Fig. 11(b) and (d) shows the sequential change of the topological graph in the *separation* and *partial occlusion* cases. In the *separation* case, the graph is separated into a moving part and a stationary part, as in Fig. 11(b) while the graph is still connected although there are occluded parts by the human hand.

The newly generated objects construct their own GMMs and make new tracks in the temporal associations with existing updated objects in MFT process. In order to apply MFT to multiple GMM-based objects, each node in the graph represents each object O and corresponding $gmm(O)$. At time t, the nodes in a new frame, such as v_{21}, v_{22}, and v_{23} in Fig. 3(a), are generated from the segmented object set $\mathscr{O}_t$ of (11) while at the previous time frames $t-d$, the matched nodes, are assigned as the identified objects from $\mathscr{T}_{t-d}$ and corresponding GMMs. The weight function between the objects is characterized by the L2-distance of the GMMs, as in (5). Because the L2-distance presents a smaller number with greater closeness of the two GMMs, the weight function is defined by (19), and takes a value between 0 and 1.

$$weight(O_1, O_2) = 1 - \frac{d_{L2}(gmm(O_1), gmm(O_2))}{\max\limits_{i,j}(d_{L2}(gmm(O_i), gmm(O_j)))}. \tag{19}$$

5. Experiments and results

The purpose of the proposed method is to track multiple objects robustly and efficiently. These two properties are in a trade-off relationship, and several parameters in the algorithm are involved in this trade-off. For example, the ratio of the original point set and the sampled point set determines the extent of data and computation time reduction, but this reduction is accompanied by a loss of information used to represent the object. In this section, the control variables and corresponding effects are investigated in terms of the tracking accuracy and the computational efficiency. We conducted several experiments to examine the following questions:

- According to the GMM representation method, how much does the constructed object model affect the tracking performance of multiple objects in situations of dynamic movements and multiple interactions?
- What is the limitation of the proposed tracking algorithm to be feasible for use in terms of the speed of objects and the number of objects?

The experiments involve tracking human hands and multiple objects on a table. The data is captured by a RGB-D camera (ASUS Xtion) established at a height of 90 cm on the table. The size of the workspace is 70 × 70 × 70 cm and the surface of the table is not included in the space. The computation device is an Intel i7 2.8 GHz CPU and RGB-D point set data, size of 640 × 480, is captured at an average of 30 Hz frequency. The data is then transformed into 6-dimensional point data (x, y, z, r, g, and b) with respect to the coordinate on the table, and data outside of the workspace is cut out. The data in the workspace is down-sampled with a given sampling distance by using VoxelGrid filter in [33], and the reduced data enter the proposed tracking process. In order to analyze the proposed algorithm, the control variables to represent the object model include sampling distance, simplification ratio, and a 3-$d(p = \{x, y, z\})$ or 6-$d(p = \{x, y, z, r, g, b\})$ GMM. The size of extension frames in MFT-GMM is also a control variable for the tracking algorithm, and the number of objects and the speed of movements are control variables of the object state. The performance of the tracking results is investigated in terms of tracking accuracy and computation time. In order to obtain ground truth data, each object has different color in the experiments, and the tracking accuracy here denotes the rate of the correctly segmented points to the total points for all frames.

$$accuracy[\%] = \left(\frac{\sum_t^T \sum_i^{N_o^t} n(p_i^t | color(p_i^t) = color(o_i^t))}{\sum_t^T \sum_i^{N_o^t} n_i^t}\right) \times 100. \tag{20}$$

The computation time was measured by calculating computed frames per second (FPS) in all frames.

5.1. Performance of tracking multiple objects in dynamic movements

In this experiment, we want to measure how much the GMM representation affects the tracking accuracy and the computational efficiency. The experiments were run with multiple moving objects in the *multiple contacts* situation. Fig. 12 shows six movements of two hands translating and rotating a white object in three dimensions. Each action was repeated five times, and took around 25 s in total. Each instance of captured point set data was reduced by down-sampling with a constant sampling distance. This experiment was performed with four different sampling distances ranging from 0.01 to 0.025 because the size of the initial GMM, n, is a substantial control parameter for the tracking accuracy and the computational efficiency. The initial GMM is then constructed with a diagonal covariance of the σ value, as in the corresponding sampling distance. Table 1 shows the details of the six experiments. Six cases were evaluated according to the changes in the values of n and m by controlling the down-sampling distance within a range of 0.01 m to 0.025 m and the simplification ratio within the range of 0.1 to 0.3, and the choice of using 3-d or 6-d GMM representation.

Fig. 13 shows the averaged tracking accuracies and computation times of the six tasks illustrated in Fig. 12. In order to find the optimal control parameters of the sampling distance and simplification ratio for each GMM representation, the requirement of the computable frames per second was set to a minimum of 5 FPS. In Fig. 13, the available parameter values of 3-d GMM are 0.025 m for the sam-

[11] The connectivity of the topological graph is tested using the connectivity test algorithm in the LEMON Graph Library (https://lemon.cs.elte.hu).

pling distance with any simplification ratio and 0.02 m for the sampling distance with a simplification ratio of less than 0.15. Among these values, the highest tracking accuracy with the computation time constraint can be obtained by the parameters of 0.02 m for the sampling distance and 0.15 for the simplification ratio, thus achieving average 91.13% accuracy. In the same way, the 6-*d* GMM representation has the optimal parameters of 0.02 m for the sampling distance and 0.15 for the simplification ratio, and these values achieve average 97.74% accuracy.

Fig. 14 shows selected snapshots of test data, rotation in the z-direction, with a sampling distance of 0.02 m and a simplification ratio of 0.15 for 6-*d* GMM. The figures in the first row are original RGB-D data. Initially, two hands and the white box are separated from each other as shown in the first columns of Fig. 14. The second to the fourth columns show the sequence of the test motions with multiple contacts between the three objects. The figures on the second row illustrate the GMM with TSTG of each object, and the final results of the proposed tracking algorithm are depicted in the figures in the third row.

5.2. Limitation of the tracking algorithm in terms of the movement speed

The objective of the second experiment is to find the limitation of the proposed method in terms of the movement speed of the object. Because it is not possible to change human movement accurately with different velocities, we performed the experiment by skipping frames alternately in the captured data of a original movement. The number of skipped frames in a series corresponds to the number of multiplication of the original speed. We used the data of the first experiment as the original speed, average 0.86 m/s for translation motion and average 2.11 rad/s for rotation motion, and calculated the tracking accuracy according to the control variables of the GMM representation with a fixed sampling distance of 0.02 m. The speed of movement was changed from two to ten times the original speed of movements.

Fig. 15 shows the tracking accuracy results of 3-*d* GMM and 6-*d* GMM according to the increasing speed of objects. We set the breaking point of the algorithm as 90% for 3-*d* GMM and 97% for 6-*d* GMM, because these are the values obtained from the first experiment. As the figures show, the both accuracies decrease as the speed increases, but 6-*d* GMM is more robust than 3-*d* GMM in terms of tracking fast moving objects. With 6-*d* GMM representation, the limitation speed of the moving objects can be considered five times than the original speed, that is 4.3 m/s for translation or 10.55 rad/s for rotation movement.

5.3. Limitation of the tracking algorithm in terms of the number of objects

The third experiment was performed to test another limitation of the algorithm in terms of the computation time according to the number of objects and the length of frames, *k* in MFT-GMM, to construct temporal associations. The search space of the extension graph in MFT-GMM, as in Fig. 3(c), increases according to the number of objects and the length of frames, which increases the computation time. In this experiment, a human incrementally carried new objects into the scene and the computation time was measured with fixed control variables for object representation (0.02 m sampling rate, 0.15 simplification ratio and 6-*d* GMM). The number of objects increased from one to eight, which caused the increase of the number of points from 20 to 650. The length of frames, *k* in MFT-GMM, varied from 30 to 150 (from one seconds to five seconds in 30 Hz) to track objects in the case of full occlusion as much as the same length of time. As Fig. 16 shows, the computation time becomes exponentially expensive according to the number of objects and points, and the larger length of frames requires more computation time as well. The real-time performance of the proposed algorithm depends on many combinations of the parameters, and the limitation of the algorithm can be determined by the given situation. For example of this experiment, with the breaking point of 5 FPS, the proposed algorithm shows the limitation of tracking around 400 points in nine objects with 30 of *k*.

5.4. Tracking multiple object in various interaction situations

Finally, the proposed algorithm was tested in a real situation of tracking multiple objects on a table including various interaction cases shown in Fig. 1. This test is not computationally analyzed but Fig. 17 shows snapshots of the results.[12] The first task aims to test the *separation* case and tracking fast moving objects. As shown in Fig. 17(a), a human hand carries a small object into the scene, and then two hands play with the object by pushing it to each other. The second task involves the *full occlusion* case. Two hands pass over the two small objects alternatively, and hold each object for a while. Fig. 17(b) shows that the fully occluded objects are recovered as soon as they appeared again with constant id number of each object. The third task tests the *partial occlusion* and multiple *contact* cases. Even though a shadow of each hand divides an object into two elements, the object preserves its points. As shown in Fig. 17(c), there are error points in the case of multiple contact situation because two contacted objects (white boxes) have similar color information.

The values of the control variables are as follows: 6D GMM, 0.02 m sampling rate, 0.15 simplification ratio, 60 frames of MFT-GMM, 10 frames of MFT-G, and 0.98 of α.

6. Conclusion and further works

In this paper, we presented a novel tracking method for multiple moving objects from RGB-D point set data. In particular, this method adopted a Gaussian mixture models (GMM) to represent any arbitrary object without prior knowledge. The flexibility of the model-free approach suffers from the dynamic movements of the objects and interaction cases such as *contact* and *occlusions* among multiple objects. The proposed method enhanced the robustness of the tracking task by suggesting a framework of incremental object modeling and multiple object tracking methods. The object model was represented by a GMM with a temporal-spatial topological graph (TSTG) and each object model can be updated at every time step. In order to estimate new measurement of each object, a GMM-based robust registration method and Maximum weighted Likelihood point-matching process were proposed. A multi-frame tracking algorithm was used to make robust temporal associations among multiple objects and among Gaussians in an object. The performance of the proposed algorithm was tested and the relation between tracking accuracy and the computational efficiency was examined by various experiments. The results showed that this method successfully attains more than 97% tracking accuracy and 5 FPS computation time with 6-*d* GMM representation. The optimal parameters were a simplification ratio of 0.15 in the cases of about 400 points at every time frame, which is reduced by down-sampling with 0.02 m sampling distance from the original point data set.

Although the results showed the feasibility of the algorithm, there are some areas that can be supplemented in further work. First, in order to enhance robustness, the GMM-based object representation should be combined with several filtering methods using the history of each track. Second, the parameter values obtained by the experiments are not truly optimal because the size of the Gaussians could not reflect the shape information of each object. The

[12] Movie clips of the tracking result of the scenario can be found at the official webpage of this work: (http://robot.kaist.ac.kr/project/pmot).

true optimal parameter values should be automatically determined by and adapted to the observed data. This will be achieved in the future work by introducing hierarchical bayesian nonparametrics [37]. Third, this study only showed the results of the proposed algorithm, but the comparisons with other feature-based methods in terms of flexibility and robustness are also required in the future work. Fourth, one of the objectives of this study is a real-time implementation. Although the algorithm could perform with 5 FPS computation speed, the size of points are limited to 400. The algorithm will be designed by using Graphical Processors (GPU) and then tracking tasks will be extended to the larger workspace. These further studies will extend the method to be used for modeling and tracking articulated objects without prior knowledge. That is, a robot can learn new objects and related skills in an unstructured environment merely by observing a human demonstration.

Acknowledgments

This work was supported in part within the DFG excellence initiative research cluster "Cognition for Technical System-CoTeSys", and financially supported by the Industrial Strategic Technology Development Program (10044009, Development of a self-improving bidirectional sustainable HRI technology) funded by the Ministry of Knowledge Economy(MKE), Korea.

References

[1] E. Aksoy, A. Abramov, J. Dörr, K. Ning, B. Dellen, F. Wörgötter, Learning the semantics of object-action relations by observation, The International Journal of Robotics Research 30 (2011) 1229–1249.
[2] B. Anderson, J. Moore, Optimal Filtering, 11, Prentice-Hall Englewood Cliffs, NJ, 1979.
[3] Y. Bar-Yosef, Y. Bistritz, Discriminative simplification of mixture models, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2011, pp. 2240–2243.
[4] A. Basu, I. Harris, N. Hjort, M. Jones, Robust and efficient estimation by minimising a density power divergence, Biometrika 85 (1998) 549–559.
[5] J. Bilmes, A gentle tutorial of the em algorithm and its application to parameter estimation for Gaussian mixture and hidden markov models, International Computer Science Institute 4 (1998) 126.
[6] N. Blodow, L. Goron, Z. Marton, D. Pangercic, T. Ruhr, M. Tenorth, M. Beetz, Autonomous semantic mapping for robots performing everyday manipulation tasks in kitchen environments, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2011, pp. 4263–4270.
[7] L. Chang, J. Smith, D. Fox, Interactive singulation of objects from a pile, in: International Conference on Robotics and Automation (ICRA), IEEE, 2012, pp. 3875–3882.
[8] H. Chui, A. Rangarajan, A feature registration framework using mixture models, in: IEEE Workshop on Mathematical Methods in Biomedical Image Analysis, IEEE, 2000, pp. 190–197.
[9] N. Dantam, I. Essa, M. Stilman, Linguistic transfer of human assembly tasks to robots, in: International Conference on Intelligent Robots and Systems (IROS), IEEE/RSJ, 2012, pp. 237–242.
[10] M. Davide, Tracking human motion with multiple cameras using articulated ICP with hard constraints, Ph.D. thesis, Dipartimento di Informatica, Universita degli Studi di Verona, 2009.
[11] A. Dempster, N. Laird, D. Rubin, Maximum likelihood from incomplete data via the em algorithm, Journal of the Royal Statistical Society Series B (Methodological) (1977) 1–38.
[12] H. Dindo, G. Schillaci, An adaptive probabilistic approach to goal-level imitation learning, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2010, pp. 4452–4457.
[13] P. Fitzpatrick, G. Metta, Grounding vision through experimental manipulation, Philosophical Transactions of the Royal Society of London Series A: Mathematical, Physical and Engineering Sciences 361 (2003) 2165–2185.
[14] V. Garcia, F. Nielsen, R. Nock, Hierarchical Gaussian mixture model, in: IEEE International Conference on Acoustics, Speech, and, Signal Processing (ICASSP), 2010.
[15] J. Goldberger, H. Greenspan, J. Dreyfuss, Simplifying mixture models using the unscented transform, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (2008) 1496–1502.
[16] J. Goldberger, S. Roweis, Hierarchical clustering of a mixture model, Advances in Neural Advances in Neural Information Processing Systems 17 (2005) 505–512.
[17] L. Greengard, X. Sun, A new version of the fast gauss transform, Documenta Mathematica 3 (1998) 575–584.
[18] S. Grigorescu, D. Pangercic, M. Beetz, 2d–3d collaborative tracking (23ct): towards stable robotic manipulation, in: IEEE-RSJ International Conference on Intelligent Robots and Systems (IROS), Workshop on Active Semantic Perception, 2012.
[19] B. Jian, B. Vemuri, A robust algorithm for point set registration using mixture of gaussians, in: 10th IEEE International Conference on Computer Vision (ICCV 2005), IEEE, 2005, pp. 1246–1251.
[20] B. Jian, B. Vemuri, Robust point set registration using gaussian mixture models, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (2011) 1633–1645.
[21] Y. Jiang, M. Lim, C. Zheng, A. Saxena, Learning to place new objects in a scene, International Journal of Robotics Research (IJRR) 31 (2012) 1021–1043.
[22] S. Knoop, S. Vacek, R. Dillmann, Sensor fusion for 3d human body tracking with an articulated 3d body model, in: IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2006, pp. 1686–1691.
[23] M. Krainin, P. Henry, X. Ren, D. Fox, Manipulator and object tracking for in-hand 3d object modeling, The International Journal of Robotics Research 30 (2011) 1311–1327.
[24] B. Lau, K. Arras, W. Burgard, Multi-model hypothesis group tracking and group size estimation, International Journal of Social Robotics 2 (2010) 19–30.
[25] Z. Liu, D. Lee, W. Sepp, Particle filter based monocular human tracking with a 3d cardbox model and a novel deterministic resampling strategy, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2011, pp. 3626–3631.
[26] M. Luber, L. Spinello, K. Arras, People tracking in rgb-d data with on-line boosted target models, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2011, pp. 3844–3849.
[27] A. Miller, P. Allen, Graspit! a versatile simulator for robotic grasping, IEEE Robotics and Automation Magazine 11 (2004) 110–122.
[28] L. Montesano, M. Lopes, A. Bernardino, J. Santos-Victor, Learning object affordances: from sensory-motor coordination to imitation, IEEE Transactions on Robotics 24 (2008) 15–26.
[29] F. Nielsen, V. Garcia, R. Nock, Simplifying gaussian mixture models via entropic quantization, in: 17th European Conference on Signal Processing (EUSIPCO), 2009.
[30] J. Prankl, M. Zillich, M. Vincze, 3d piecewise planar object model for robotics manipulation, in: IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2011, pp. 1784–1790.
[31] S. Rusinkiewicz, M. Levoy, Efficient variants of the icp algorithm, in: Third International Conference on 3-D Digital Imaging and Modeling, Proceedings, IEEE, 2001, pp. 145–152.
[32] R. Rusu, N. Blodow, Z. Marton, M. Beetz, Close-range scene segmentation and reconstruction of 3d point cloud maps for mobile manipulation in domestic environments, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2009), 2009, pp. 1–6.
[33] R. Rusu, S. Cousins, 3d is here: point cloud library (pcl), in: IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2011, pp. 1–4.
[34] R. Schnabel, R.Wahl, R. Klein, Efficient ransac for point-cloud shape detection, in: Computer Graphics Forum, vol. 26, Wiley Online Library, 2007, pp. 214–226.
[35] D. Schulz, W. Burgard, D. Fox, A. Cremers, People tracking with mobile robots using sample-based joint probabilistic data association filters, The International Journal of Robotics Research 22 (2003) 99–116.
[36] K. Shafique, M. Shah, A noniterative greedy algorithm for multiframe point correspondence, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2005) 51–65.
[37] Y.W. Teh, M.I. Jordan, Hierarchical Bayesian nonparametric models with applications, in: N. Hjort, C. Holmes, P. Muller, S. Walker (Eds.), Bayesian Nonparametrics: Principles and Practices, 2010, pp. 158–207.
[38] Y. Tsin, T. Kanade, A correlation-based approach to robust point set registration, Computer Vision-ECCV 2004 (2004) 558–569.
[39] R. Ueda, Point clouds library: Tracking.
[40] C. Veenman, M. Reinders, E. Backer, Resolving motion correspondence for densely moving points, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2001) 54–72.
[41] K. Zhang, J. Kwok, Simplifying mixture models through function approximation, IEEE Transactions on Neural Networks 21 (2010) 644–658.
[42] L. Zhang, J. Sturm, D. Cremers, D. Lee, Real-time human motion tracking using multiple depth cameras, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2010.

Please cite this article in press as: S. Koo et al., Incremental object learning and robust tracking of multiple objects from RGB-D point set data, J. Vis. Commun. (2013), http://dx.doi.org/10.1016/j.jvcir.2013.03.020