Human Body Part Multicontact Recognition and Detection Methodology

Kwan Suk Kim[1], Luis Sentis[1]

Abstract— **In this paper we focus on a mobile platform which physically interacts with a human operator. We detect the contact gestures of a human operator in real-time using a lab-made time-of-flight 3D scanner mounted on the platform as well as rotary torque sensors mounted along the drivetrain of its omni-directional wheels. Through the fusion of these two different sensors, touch gestures of an operator are processed inferring information about the body parts in contact and the applied forces. Behaviors that respond to touch-based gestures are programmed a priori, and with the previous sensor data we classify them into a set of known contact gestures that allow the platform to quickly react. We investigate these physical human-robot cooperative functions in a testbed consisting of a sensorized mobile platform and a human operator.**

I. Introduction

In this paper, we study physical Human Robot Interaction (pHRI) in personal robotic platforms using human contact detection and contact gesture queues. Previously, to interact with robots, various methods have been devised such as those relying on dedicated input devices, body language visual recognition, voice recognition, and sensorized skins or touch devices [17] among others. In our study, we focus on fusing visual recognition with contact sensing on mobile platforms, which allows close contact with people. Existing work on fusing visual and contact recognition [9] has primarily focused on robotic manipulators and on using external structured light 3D sensing. However, this type of method does not directly apply to omnidirectional mobile platforms like our studies. First mobile platforms require different dynamical models that incorporate traction and roller constraints as well as models of roller friction. Second, mobile platforms need to carry the 3D sensor on board, and therefore time-of-flight 3D sensing is more suitable for close range detection. Third previous methods have focused on detecting contacts and the corresponding forces. In addition to these capabilities, we focus on inferring which human body parts are in contact with the robot. Ultimately, these capabilities endow richer pHRI behaviors. Finally, previous work on multicontact detection has focused on detection only, but not on multicontact gesture communications as explored here. As such, the aim of this work is to enable mobile ground platforms to physically interact with humans by means of multicontact gestures.

Ultimately, if robots are to be used as personal companions for boosting our comfort and productivity, we believe they will benefit by these type of close contact capabilities. The goal is to increase the communication modalities with robots to become more intuitive by exploiting contact interactions.

One type of intuitive method for HRI is based on using body language. A popular method for gesture recognition is using depth image data made from structure light such as the Kinect^{TM} sensor. One of the drawback is that this type of sensor is limited to indoor environments because the infrared light (IR) is vulnerable to sunlight. An alternative to structure light sensing is a laser scanner based on time-of-flight measurements. In our case we use this optical sensing modality because it allows to operate at close ranges and also in outdoor environments. To leverage gesture recognition to HRI in robot companion applications, we focus both on the detection phase but also on the identification of the human body parts approaching or in contact with the robotic platform. Such capability allows to closely analyze the nature of the intended behaviors.

Additionally touching has became pervasive for handheld devices. We feel that our technology will allow to interact with robots in similar ways, enabling complex touch behaviors. In our case, we feel that the possibilities are enormous as many body parts can be used for communication and the entire robot structure can be touched.

One approach to detect contact is to use a vision system recognizing contact with nearby object [4]. Also, tactile sensors on robot skins are popular to detect a contact. In such case, tactile sensors are attached to an outer skin of the robot. The problem is that all the exterior of the robot needs to be covered by the sensorized skin to guarantee whole-body contact detection. Another limitation is that touch skins cannot recognize human body parts. Lastly, joint torques on robotic manipulators have been used to infer contact information [7]. Usually, sensing contact forces has enabled some level of safety during accidental collisions [5]. Previously, we have investigated whole-body contact detection on mobile platforms to provide safety [6], but not for human intention recognition. Since a contact force is just one part of touch-based gestures, we cannot fully estimate the human intention solely based on forces. To push the boundaries of physical HRI, we suggest a multi-contact gesture recognition method with human body-part awareness. Our contact-based gesture recognition method allows differentiation between intentional gestures and un-intentional human activity. The reason is that the robot can recognize what contacts are being made and match them against the human body parts in contact. This allows to be precise on the detection of intention. Compared to other input devices, a physical contact interaction can be more intuitive for a certain class of communication queues. It could

This paper is supported by ONR N00014-15-1-2507 and NASA NRI NX12AM03G. [1]The authors are with the Department of Mechanical Engineering, University of Texas at Austin, Austin, TX 78712, USA kskim@utexas.edu, lsentis@utexas.edu

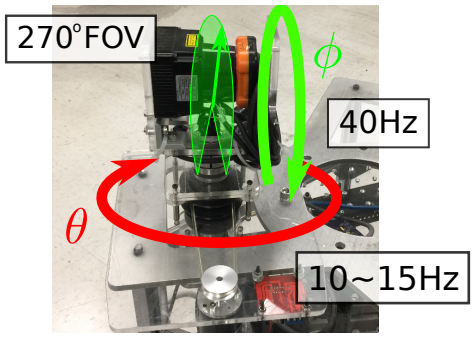

Fig. 1. **3D scanner** made from a 2D LIDAR, Hokuyo UTM-30LX, can scan all around the robot. While the 2D lidar scans a vertical plane (green plane), the rotating gimbal rotates the plane with $10 \sim 15$Hz speed.

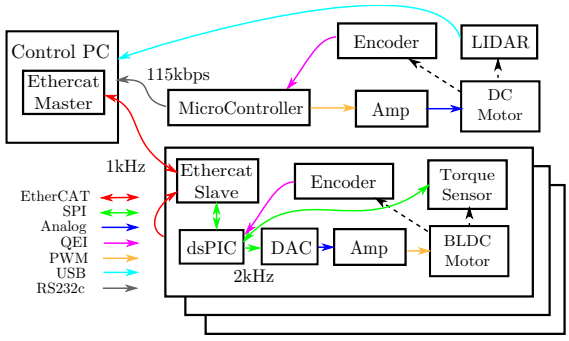

Fig. 2. **The electrical system of the mobile platform** consists of actuation parts and the 3D scanner. The microcontroller, which is Raspberry PI 2, for the 3D scanner controls the rotational speed of the gimbal and measures the orientation of the laser ray. It generates the orientation from the synchronization signal from the 2D LIDAR and the encoder on the gimbal.

ultimately relief operators from getting distracted during their interaction with personal mobile platforms.

In our approach, a human behavior is estimated by a depth image generated by a rotary laser scanner. The estimated gesture is not immediately recognized as an intentional command. When an external force is detected by the mobile platform's torque sensors, contact gestures can be recognized from the behavior information of the human. As a response, the robot can trigger a behavior that services the estimated intention.

II. Related Works

To detect humans and objects in unstructured environments, exteroceptive sensors such as cameras and range finders are often used. One of them is the Kinect sensor [2]. The original application of this structured light sensor was as a gaming input device, so the sensor did not require 360° scanning capabilities and had a relatively narrow field of view. Additionally, the Kinect and similar sensors have a relatively large minimum focal length making them unable to detect proximity at close range. Recently, [9] detected multicontact on a robotic manipulator using a Kinect sensor. The sensor was located outside of the manipulator to allow it to work at the prescribed operating distances. This placement constraint highlights that Kinect type of sensors cannot be used to detect close range contacts or proximity in mobile platform compared to the proposed system which has with the minimum distance of 10cm. Video cameras can also be used to detect humans around robots, but their narrow field of view limits their effectiveness. Additionally, video cameras have very noisy and low resolution depth sensing making them less suitable to detect proximity behaviors. A third method consists on using a laser range finder. A large number of research on human detection with these types of devices focus on finding and tracking pedestrians with 2D scanning [13] which is used in autonomous cars to avoid collisions, for instance. To scan a 3D environment with a range finder, multiple laser rays are shot and multiple planes are scanned simultaneously by a multilayer 3D LIDAR [15] or multiple 2D LIDARs [18]. On the other hand, some researchers reconstruct a 3D environment with a single LIDAR by rotating it over its axis [13]. 2D LIDARs on rotary mounts are embedded in quadrotors [3], on mobile robots [12], or on handheld poles [10]. There are several types of 3D scanning methods addressing the rotating axes of the LIDAR and the rotary mount [16]. In [13], the authors track human behavior via HMM's base on a planar LIDAR scanning. In [1], the authors suggested combining an interlaced 3D scanner with a 2D planar LIDAR. This scanner is also known as Lissajous scanner, and is used for multiresolution microscopes [14]. In [4], the authors attempted to find a contact based on video sequences, but their estimated contacts did not include magnitude and direction of the applied forces.

III. Hardware

A. Omni-directional Mobile Robot

Trikey shown in Figs. 7, 8 is a holonomic omnidirectional mobile robot which has torque sensors on its drivetrain [6]. The external forces applied by users are detected by the torque sensors via model based whole-body sensing algorithms. We can estimate the location, magnitude and direction of external forces and collisions.

B. Interlaced Scanning

We implement a 3D scanner by employing a 2D LIDAR (Hokuyo UTM-30LX) on top of a gimbal as shown in Figs. 1, 2. The LIDAR triggers a synchronization signal to the GPIO port of the microcontroller. Whenever the laser in the sensor rotates once with 40Hz and the LIDAR generates a sequence of distance, we interpolate the timestamps of the signals, and generate a sequence of a tuple which consists of a timestamp, angle, and distance. Also, we attach an 2500-CPS optical encoder from US Digital Inc. (E6-2500-1000-IE-S-H-D-B) to the gimbal, and the QEI signals from the encoder are also fed into the GPIO ports. To deal with the signals from the LIDAR and the encoder, we execute a sequential program on the microcontroller, and its loop period is 300kHz. By merging the signals, we generate the orientation of the laser ray and deliver it to the control PC through RS232c serial communication with 115kbps baudrate. The gimbal actuator is powered by a 12V DC motor which is controlled by

L298N DC motor driver. The rotational speed is controlled by PWM signals and kept between 10 and 15Hz. There are two rotational axes in the 3D scanner. ϕ is the angle of the laser ray in the 2D LIDAR, and θ is that of the gimbal rotating the 2D LIDAR as shown in Fig 1. When the laser ray shot by the 2D LIDAR collides with an object, the collision point with distance d can be expressed in Cartesian coordinate as follows.

$$\mathbf{p} = \begin{bmatrix} d\cos\theta\cos\phi & d\sin\theta\cos\phi & d\sin\phi \end{bmatrix}^T \quad (1)$$

The scanned 3D position, $\mathbf{p}$, is specified by three variables d, ϕ, and θ, which represent polar coordinates. There are two different types of scanning methods: progressive (raster) scanning and interlaced (Lissajous) scanning [14], [1]. Progressive scanning methods are implemented in most of the 3D scanners made from 2D LIDARs. Typically, the rotation of the gimbal is much slower than the 2D scanning, and the points captured during the previous rotation are replaced with the new points. Therefore, the points in the constructed point cloud from the progressive scan have high correlation between position and time because the scanning is conducted sequentially from one scan line to another scan line. However, in the case of the interlaced scanning, the scanning speed of the gimbal is comparable to the LIDAR scanning which means the scanning method is more suitable for tracking objects moving fast because the interlaced scanning is more responsive than the progressive scanning, and the points clouds are generated from the points captured during several rotations of the gimbal. Considering the point cloud generated from the progressive scanning, the temporal sequence of the points is usually ignored meaning that we get a snapshot taken at a given time. On the other hand, the points from interlaced scanners have a low correlation between position and time because adjacent points can belong to different scan lines. Therefore, each point has its own generation time which is not related to the position, so we need to deal with not only position of the point but also its timestamp.

IV. Contact Gesture Recognition

An external force estimation method has been derived with certain limitations as described in [6]. With the help of a 3D scanner, we can also identify the location of the contact which generates an external force and, relax some of the constraints of the estimation. In addition to the estimated contact forces, the point clouds generated by the 3D scanner include the information about the object making the contact. We assume that the object is a human body, and identify which parts of the human body make contacts with the mobile platform. All the estimated contact information is used to generate a contact gesture.

A. Point Cloud Registration

When the LIDAR sensor and the encoder of the gimbal system generate a polar coordinate of the shooting laser, it occupies one voxel in an octree [11] with the resolution of $2\text{cm}\times2\text{cm}\times2\text{cm}$. We register the shape of the top plate of the mobile platform as a triangular plane, and voxels outside of the plane are considered to be separate objects. So far, all the voxels in the space are separated into two groups: platform voxels and object voxels. To identify whether there is an object making a contact with the platform, we measure the distance from each object voxel and the triangular plane, and if the distance is less than a given threshold, we identify the voxel as just next to the platform. Fig. 4 shows the platform voxels and the object voxels which are making contacts with the platform. The object voxels are grouped together, and the mean position of the voxels in each group corresponds to the location of the contact which is used in the subsequent section.

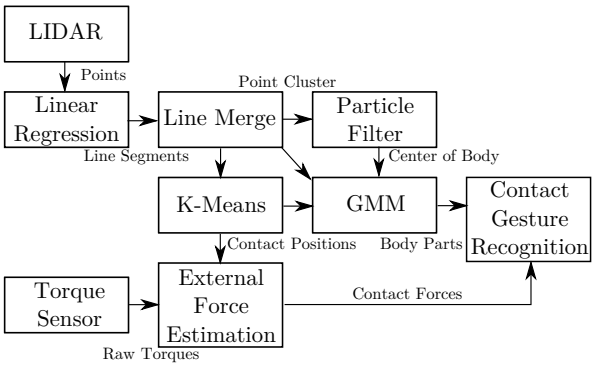

Fig. 3. **Contact Gesture Estimation** is implemented using the 3D scanner and the torque sensors. The estimated contact gesture consists of the number of contacts, the location and the force vector of each contact, and the human body part that makes the contact.

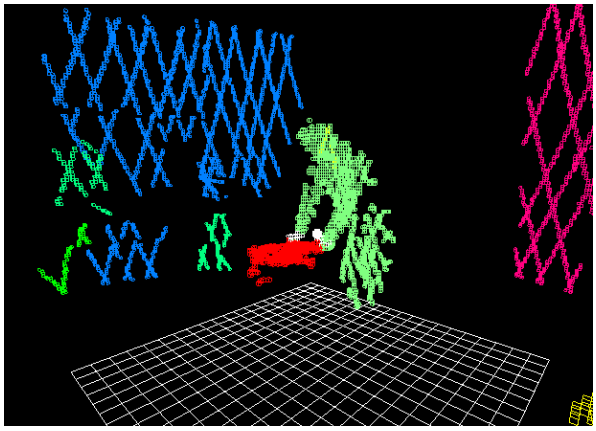

Fig. 4. **An occupancy map evaluated by Octree** includes the mobile platform and nearby objects. The mobile platform (red voxels) is identified from the predefined shape.

B. Point Clustering and Object Tracking

After the points are generated from the 2D LIDAR scanning and added to the octree, the points are classified into several point clusters. To reduce the computation time for the clustering, we form a line segment from the LIDAR data using the incremental least square linear regression algorithm, and each line segment is classified into the nearest point cluster. Then, we can deal with each point cluster as an object. Some objects can be human bodies and others

are environments. In this paper, we assume that any objects close to the robot are human operators who want to interact physically with the robot.

To estimate the contact gesture, we start by tracking human body which results in a set of time trajectories of the human operator. To incorporate both the hitting laser and the missing laser, we use a particle filter for the object tracking, and each particle represents the candidate center point of the object. We assume that the shape of the tracking object is an ellipsoid which can be expressed with the covariance matrix of the point cluster distribution, $\mathbf{\Sigma}$, as follows.

$$(\mathbf{x} - \mathbf{x}_c)^T \mathbf{\Sigma}^{-1} (\mathbf{x} - \mathbf{x}_c) = 1 \tag{2}$$

where $\mathbf{x}_c$ is the center of the object. If the j-th laser ray from the 3D scanner hits a point, z_j, then we can test whether the point is located in the object whose center is the i-th particle through the weighted distance, d_{ij} between them as follows.

$$d_{ij} = (\mathbf{x}_i - \mathbf{z}_j)^T \mathbf{\Sigma}^{-1} (\mathbf{x}_i - \mathbf{z}_j) \tag{3}$$

where $\mathbf{x}_i$ is the postion of the i-th particle. If d_{ij} is smaller than or equal to 1, that means the point belongs to the object. Therefore, we can generate a conditional probability of the observation where the j-th laser ray hits on $\mathbf{z}_j$ if the center of the object is located on the i-th particle, $\mathbf{x}_i$ as follows.

$$P(\mathbf{z}_j | \mathbf{x}_i) = \begin{cases} \epsilon & : \ d_{ij} \leq 1 \\ 0 & : \ d_{ij} > 1 \end{cases} \tag{4}$$

Then, the conditional probability that the center of the object is located on the i-th particle $\mathbf{x}_i$ given the observation, $\mathbf{z}_j$ can be derived from the Bayesian rule as follows.

$$P(\mathbf{x}_i | \mathbf{z}_j) = \frac{P(\mathbf{x}_i) P(\mathbf{z}_j | \mathbf{x}_i)}{P(\mathbf{z}_j)} \tag{5}$$

where

$$P(\mathbf{z}_j) = \sum_{\mathbf{x}_i, d_{ij} \leq 1} \epsilon P(\mathbf{x}_i) \tag{6}$$

When the laser misses the object, the missed laser ray can also be used to confirm that the object does not exist on the ray. The laser ray can be represented as a matrix equation $\mathbf{A}\,\mathbf{z} = \mathbf{0}$ because the laser comes from the origin. Then, the conditional probability of the missed laser ray given the object position, $\mathbf{x}_i$ can be expressed as follows.

$$P(\mathbf{A}\,\mathbf{z} = \mathbf{0} | \mathbf{x}_i) = \begin{cases} \eta & : \min_{\mathbf{z}_j \in \mathbf{z}} d_{ij} > 1 \\ 0 & : \min_{\mathbf{z}_j \in \mathbf{z}} d_{ij} \leq 1 \end{cases} \tag{7}$$

The conditional probability of the i-th particle with respect to the given missed laser ray events, $P(\mathbf{x}_i | \mathbf{A}\,\mathbf{z} = \mathbf{0})$ also can be derived by the Bayesian rule similar to Eq. (6).

C. Contact Position and Human Posture Estimation

From the previous section, the robot can generate the point cluster of a human body and identify its center position. As described in Sec. IV-A, when the point cloud is registered in the occupancy map, we check how close each point is to the predefined mobile platform shape. Therefore, we can build a set of all the points close enough to the robot by selecting points whose distance to the platform is smaller than a given threshold, and name it as a contact point set. The points in the set are clustered by k-means algorithm [8], then the number of contacts the human body makes can be identified by the number of the clusters. The initial states of the clusters are established from the center of the points in the set, and all the points are classified into one of the clusters after the algorithm converges. Cluster with no points are removed, and the remaining clusters are considered as contact positions. In this paper, $k = 3$ which means we assume that the contact points can be both hands and the body of the human operator. By identifying contact positions, it can be assumed that the external forces are applied on the contact positions. The locations of the contacts are fed into the external force estimator as described in Sec. IV-D.

The object point cloud, the center of the cloud, and the contact positions to the mobile platform are estimated from the 3D scanner. The body posture of the human operator can also be estimated from the point cloud, then we can identify how the human operator makes a contact with the mobile platform which we call a contact gesture. To determine body posture from the point cloud, we express a human body as a Gaussian mixture model (GMM) with expectation-maximization (EM) algorithm. Each Gaussian distribution of GMM represents a human body part, and we can identify the contact gesture from the locations of the Gaussian distributions. In the paper, we use 4 mixture components which represent left and right arms, a upper body, and a lower body if there are two contact locations. If there is only one contact location, we use 3 mixture components. The initial distributions of the mixture components are initialized with the center of the point cluster and the contact locations, and the EM algorithm iterates until the change of distributions is below a given threshold. Even though the human body posture is oversimplified as only 3 or 4 mixture components, the GMM representation has enough information to identify which part of human body makes a contact with the mobile platform. With the help of estimation methods described in this section, we can track a human operator around the mobile platform, and when the operator makes a contact, we can identify which parts of the operator make contacts with which parts of the mobile platform. The whole estimation process is depicted in Fig. 3.

D. Multicontact Force Estimation

In the previous section, we have figured out where the contacts happen on the mobile platform, which are described as 3D coordinates on the robot's frame. Assume that there are N objects making contacts with the top of the platform, and the location of the i-th contact location is (x_i, y_i) which is in the local frame of the mobile platform, and the contact force is $\mathbf{F}_i = (F_{i,x}, F_{i,y})$ which includes no torque. Then, the sum of all the forces satisfies the following condition with respect to the net force and torque on its center, $\mathbf{F}_{ext} = \begin{pmatrix} F_{ext,x} & F_{ext,y} & \tau_{ext} \end{pmatrix}^T$. which can be expressed

as the following matrix form.

$$\mathbf{F}_{ext} = \mathbf{H}_N \, \mathbf{F}_N \tag{8}$$

where

$$\mathbf{H}_N = \begin{pmatrix} \mathbf{I}_{2\times 2} & \cdots & \mathbf{I}_{2\times 2} \\ -y_1 \quad x_1 & & -y_N \quad x_N \end{pmatrix} \in \mathbb{R}^{3\times 2N} \tag{9}$$

$$\mathbf{F}_N = \begin{pmatrix} F_{1,x} & F_{1,y} & \cdots & F_{N,x} & F_{N,y} \end{pmatrix}^T \tag{10}$$

From [6], the external net force on the center of the mobile platform, $\mathbf{F}_{ext}$ can be derived from the joint torques as follows.

$$\mathbf{F}_{ext} = \mathbf{J}_{c,w}^T \left(\mathbf{\Gamma}\big|_{\mathbf{F}_{ext}=0} - \mathbf{\Gamma}_s \right) \tag{11}$$

The size of $\mathbf{H}_N$ is determined by the number of contacts, and the estimation can be either overdetermined or underdetermined. In either case, we can estimate the contact forces, $\widetilde{\mathbf{F}}_N$ as follows.

$$\widetilde{\mathbf{F}}_N = \mathbf{H}_N^T \left(\mathbf{H}_N \mathbf{H}_N^T \right)^+ \mathbf{F}_{ext} \tag{12}$$

where $(\cdot)^+$ is a pseudoinverse. The solution is equivalent to a minimum norm estimation if it is underdetermined and a least mean square error estimation if overdetermined.

When there are more than one contact point, the estimation is underdetermined because the joint torque sensors cannot sense all the contact forces such as squeezing and stretching, which means the estimated contact forces, $\widetilde{\mathbf{F}}_N$ are different from the actual contact forces, $\mathbf{F}_N$. However, the multi-contact may happen simultaneously, so the estimated first contact information can be exploited during a multicontact estimation. For the first contact, we can determine the unit vector of the force, $u_1 = \begin{pmatrix} u_{1,x} & u_{1,y} \end{pmatrix}^T$ and apply it to the minimum norm estimation process as follows.

$$\mathbf{H}' = \begin{pmatrix} & \mathbf{H}_N & & \\ u_{1,y} & -u_{1,x} & 0 & \cdots \quad 0 \end{pmatrix} \in \mathbb{R}^{(3+1)\times 2N}$$

$$\mathbf{F}'_{ext} = \begin{pmatrix} \mathbf{F}_{ext} \\ 0 \end{pmatrix} \tag{13}$$

Using Eq. (13), contact forces are estimated with the first contact information as follows.

$$\widetilde{\mathbf{F}}'_N = \mathbf{H}'^T \left(\mathbf{H}' \mathbf{H}'^T \right)^+ \mathbf{F}'_{ext} \tag{14}$$

E. Reaction to Human Intention

Using a new algorithm shown in Fig. 3, we identify all contacts on the mobile platform and corresponding human body parts in contact. Subsequently, the mobile platform can respond according to the estimated contact gestures. Inspired by touch-based APIs in mobile devices, we define a multi-touch event including location of a contact, contact force vector, and human body part in contact. When the mobile platform detects n contacts, each contact is labeled with the corresponding human body part, $i \in \mathbf{P} \triangleq \{left\_hand,\ right\_hand,\ body\}$, and touch event, $t_{e,i}$ belonging to a set of multiple touches, $\mathbf{T}_e$. Each touch event includes a location vector ($\mathbf{l}_{e,i}$) and the force vector corresponding to the touch ($\mathbf{f}_{e,i}$). To react to the contact gestures, a command set, $\mathbf{C}$ is defined. Each command, $c \in \mathbf{C}$ includes a set of triggering touch information, $\mathbf{T}_c$. A triggering touch made by the human body part, i, is denoted as $t_{c,i} \in \mathbf{T}_c$, and it consists of the location vector ($\mathbf{l}_{c,i}$) and the touch force vector ($\mathbf{f}_{c,i}$). Given the estimated touches, we can find the desired command, c, from the command set from the following equation.

$$\underset{c\in\mathbf{C}}{\operatorname{argmin}} \sum_{i\in\mathbf{P}} \mu_i \left(\mathbf{T}_e,\ \mathbf{T}_c \right) \left(w_l \left| \mathbf{l}_{e,i} - \mathbf{l}_{c,i} \right| + w_f \left| \mathbf{f}_{e,i} - \mathbf{f}_{c,i} \right| \right) \tag{15}$$

where w_l and w_f are weights for distance and torque, respectively, and μ_i is a function of sets which has the following property

$$\mu_i \left(\mathbf{T}_1, \mathbf{T}_2 \right) = \begin{cases} 1, & \exists\ t_{1,i} \in \mathbf{T}_1,\ \text{and } \exists\ \mathrm{t_{2,i}} \in \mathbf{T}_2 \\ 0, & t_{1,i} \notin \mathbf{T}_1,\ \text{and } \mathrm{t_{2,i}} \notin \mathbf{T}_2 \\ \infty, & \text{otherwise} \end{cases} \tag{16}$$

V. Experiments

In this section, we conduct experiments with the mobile platform, and prove that the proposed algorithm guarantee the effective retrieval of the contact gestures. The experiments consist of 1) contact force estimation experiment in which contact positions are identified by the 3D scanner, and we estimate the contact force on each position; 2) contact gesture recognition experiment in which the posture of a human operator is estimated, and we figure out which parts of the operator make contacts with the mobile platform; and 3) proof of concept experiment in which we show examples of how the proposed contact gesture recognition can be used as a physical HRI tool.

A. Contact Force Estimation

In this experiment, we determine contact forces when a human operator makes contacts with the mobile platform. To generate the calibrated contact forces, the human operator applies 10N of contact forces by pulling the mobile platform with spring scales as shown in Figs. 5-(a)∼(c). With the 3D scanner, the contact positions are observed, and the contact forces are estimated from the contact positions and the net force generated from the rotary torque sensors.

Figs. 5-(a)∼(c) show the configurations of the contact positions. In the case of Figs. 5-(a) and (c), the contact positions are identical, and there is no difference in the 3D scanner data. However, the operator applies different contact forces, and the contact gesture recognition method is able to resolve for the different forces. In Figs. 5-(a) and (b), the contact forces are applied in the same direction, while in Fig. 5-(c)the contact forces are perpendicular. In Figs. 5-(d)∼(f), the occupancy maps are generated from the point clouds of the configurations of Figs. 5-(a)∼(c), respectively. The red voxels are the mobile platform, and the blue voxels are the human operator. The white voxels belong to the human operator, and are close enough to be considered as making a contact with the mobile platform. The pink sphere is the estimated center of the human operator from the particle filter

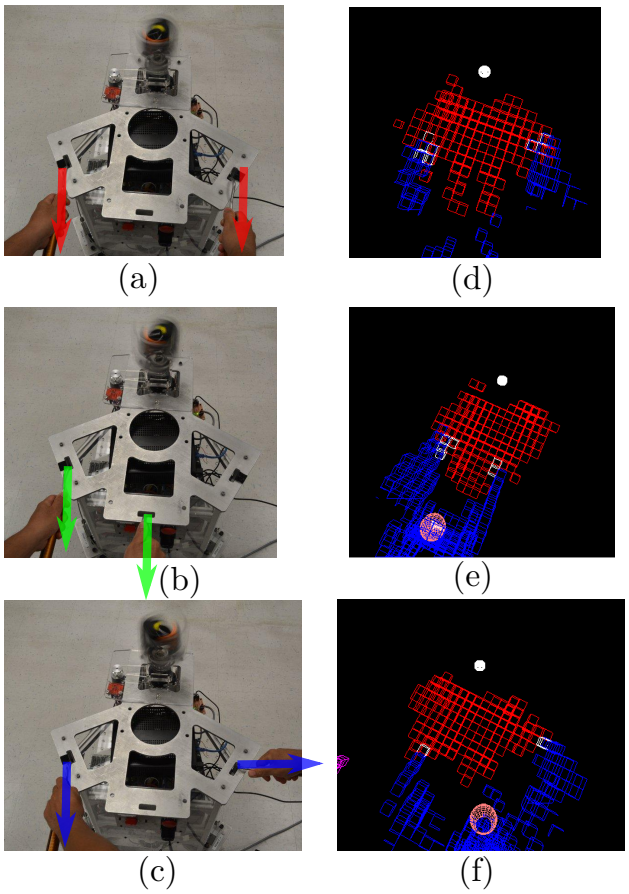

Fig. 5. **Multicontact Force Estimation** is conducted on the top of the mobile platform. Both hands of the human operator make contacts and apply some forces in (a)∼(c). In (d)∼(f), the corresponding occupancy maps are shown. The red voxels are the mobile platform, the blue voxels are the human operator, and the white voxels are the voxels of the human operator which are making contacts.

in Sec. IV-B. As described before, the contacts in Figs. 5-(a) and (c) are identical in position.

Figs. 5-(d)∼(f) show both the estimated contact positions and the contact forces. The contact positions are calculated from the average of contact voxels in the occupancy maps, and the contact forces are estimated from the net force and the contact positions with Eq. (12). Because the applied contact forces in Figs. 6-(a) and (c) are in the same direction, the estimated fores are identical from the minimum norm estimation process. On the other hand, in Fig. 6-(c), the forces are not in the same direction, and therefore some components cancel each other out, resulting in the estimated forces being different from the actual ones. To estimate contacts more precisely, the human operator makes contacts sequentially, and the information from the first contact is used as a prior. The left hand makes a contact earlier than the right hand, the direction of the first contact is added to $\mathbf{H}'$ in Eq. (13), and the contact forces are estimated with Eq. (14). The estimation with the sequential contacts in Fig. 6-(d) shows that the estimated forces are close to the actual forces.

TABLE I
COMMAND SET

Name	Part	Location (m)	Force (N)	Action
Collide	body	(0,0)	(5,0)	Move away quickly
Push	right	(0,-0.3)	(5,0)	Go straight
Pull	right	(0,-0.3)	(-5,0)	Go straight and come back
Rotate	left	(0,-0.3)	(5,0)	Rotate
	right	(0,0.3)	(-5,0)	

B. Contact Gesture Recognition

In addition to the previous experiment, we estimate the body posture of the human operator making contacts with the mobile platform. In this experiment, the following body parts of the human operator make contact: 1) a single hand, 2) both hands, and 3) a thigh. We simplify the human body and express it as three or four parts: one or two arms, an upper body, and a lower body. The number of arms is determined by the number of contacts which is identified by the k-means method. Also, while making contacts, the human operator applies forces by pulling or pushing. Fig. 7 shows the experimental results.

In Figs. 7-(a)∼(f), the human operator's posture is rendered, and the estimated contact gestures are shown in Figs. 7-(g)∼(l). In those figures, the white spheres are the estimated contact positions identified from the point clouds, and the green octahedron shows the Gaussian distribution of the estimated human body parts. Each vertex of the octahedron is 1-σ boundary of the distribution. The yellow arrows are the estimated contact forces.

In Figs. 7-(a) and (b), single pulling and pushing forces are applied to the mobile platform, respectively, and The green arrows show the applied forces. As shown in Figs. 7-(g) and (h), a single contact point is found, and the three Gaussian distributions are identified as the human body parts. The arm clusters converge to the real arm distributions, and the contact is determined to be at the end of the arms. Thus, we can determined what hands make contacts with the mobile platform. Also, the contact forces estimated from the external force estimator in Fig. 3 have the correct directions with respect to the applied forces.

In the case of the multicontact experiments shown in Figs. 7-(c) and (d), a pushing and twisting forces are applied to the mobile platform, respectively. The k-means algorithm identifies that there are two contact points, and therefore four Gaussian distributions are used for all contact situations. Both arm clusters converge towards the actual visualized arms, allowing to determine that both hands make contacts with the mobile platform. Figs. 7-(i) and (j) show the estimated forces with the contact gestures, and the directions of the contact forces are identical to the actual forces.

Contact with the lower human body is tested in Fig. 7-(e) and (f). Typically, this kind of contact means a collision that needs to be avoided. In the experiment, the human operator leans toward the mobile platform, and a pushing force is

accidentally applied to it. As shown in Fig. 7-(k) and (l), all three distributions are located on the lower human body area. By comparing the covariance of the Gaussian distributions to those of other contact events, we can distinguish between lower body contacts and hand contacts.

C. pHRI through Contact Gesture

In the last experiment, we prove the concept of HRI with the proposed contact-based gesture recognition. For the experiments, we define four commands in Table I. If the magnitude of the estimated contact force is greater than a triggering threshold of $5\mathrm{N}$, the mobile platform determines one command in the set which has the smallest test value from Eq. (15), and executes the predefined action. Fig 8 shows the operations of the mobile platform commanded through the recognized contact gesture. Even though the difference between the gestures in Fig. 8-(a) and (b) are insignificant, the commands are different by the estimated forces. Also, by identifying the human body part in contact, we can differentiate an intentional action from an undesired collision as shown in Fig. 8-(a) and (d).

VI. Conclusion

In this paper, we have devised a methodology for identifying contact gestures between humans and omnidirectional mobile platforms. To estimate contact gestures, we combine data from a 3D scanner which is constructed using a rotating 2D LIDAR and rotary torque sensors on the platform's drivetrains. We use this infrastructure to determine which human body parts make contact with mobile platforms and how much forces they apply to it. To achieve responsive and omnidirectional contact detection using the 3D scanner, we choose an interlaced scanning procedure, where its meshlike scan map enables instantaneous contact detection. Even though it is hard to reconstruct a sophisticated 3D environment with the coarse scanning sensor, our method can obtain enough contact information for effective physical HRI. We assume that all the nearby objects are human operators, but this limitation can be relaxed by adopting pervasive object classification methods. The contact information includes the location of a contact and the human body parts making contacts, which are determined via clustering methods. Also, by fusing these contact information with the rotary torque sensory data, we can estimate the contact force on each contact location. The estimation is underdetermined, so we apply a minimum norm estimation and prior contact information. Finally, we demonstrate the possibility of using contact gestures as a physical HRI tool through various proof of concept experiments. In those experiment, the mobile platform identifies the predefined contact command queues and response according to the commanded gestures. In the future, we will focus on achieving higher accuracy on detection and faster responsiveness to human gestures. The complexity of our detection algorithm is proportional to the number of nearby human operators, so it can be extended without a great effort. We will also focus on developing a more meaningful language for contact based communications for effective pHRI.

References

[1] J Wesley Anderson and Garrett M Clayton. Lissajous-like scan pattern for a gimballed LIDAR. In *Advanced Intelligent Mechatronics (AIM), 2014 IEEE/ASME International Conference on*, volume 1, pages 1171–1176, 2014.

[2] Lulu Chen, Hong Wei, and James Ferryman. A survey of human motion analysis using depth imagery. *Pattern Recognition Letters*, 34(15):1995–2006, 2013.

[3] David Droeschel, Jorg Stuckler, and Sven Behnke. Local multi-resolution representation for 6d motion estimation and mapping with a continuously rotating 3d laser scanner. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 5221–5226. IEEE, 2014.

[4] Dirk M Ebert and Dominik D Henrich. Safe human-robot-cooperation: Image-based collision detection for industrial robots. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference On*, volume 2, pages 1826–1831. IEEE, 2002.

[5] S. Haddadin, A. Albu-Schaffer, and G. Hirzinger. Requirements for Safe Robots: Measurements, Analysis and New Insights. *The International Journal of Robotics Research*, 28(11-12):1507–1527, August 2009.

[6] Kwan Suk Kim, Travis Llado, and Luis Sentis. Full-body collision detection and reaction with omnidirectional mobile platforms: A step towards safe human-robot interaction. *Autonomous Robots*, 40(2):325–341, 2016.

[7] A. De Luca. Collision detection and safe reaction with the DLR-III lightweight manipulator arm. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1623–1630, October 2006.

[8] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

[9] Emanuele Magrini, Fabrizio Flacco, and Alessandro De Luca. Control of generalized contact motion and force in physical human-robot interaction. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 2298–2304. IEEE, 2015.

[10] Mitsuhiro Matsumoto and Shin'ichi Yuta. 3d sokuiki sensor module with roundly swinging mechanism for taking wide-field range and reflection intensity image in high speed. In *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*, pages 349–353. IEEE, 2011.

[11] Donald JR Meagher. *Octree encoding: A new technique for the representation, manipulation and display of arbitrary 3-d objects by computer*. Electrical and Systems Engineering Department Rensseiaer Polytechnic Institute Image Processing Laboratory, 1980.

[12] Kazunori Ohno, Toyokazu Kawahara, and Satoshi Tadokoro. Development of 3d laser scanner for measuring uniform and dense 3d shapes of static objects in dynamic environment. In *Robotics and Biomimetics, 2008. ROBIO 2008. IEEE International Conference on*, pages 2161–2167. IEEE, 2009.

[13] Anand Panangadan, Maja Matarić, and Gaurav S Sukhatme. Tracking and modeling of human activity using laser rangefinders. *International Journal of Social Robotics*, 2(1):95–107, 2010.

[14] Tomas Tuma, John Lygeros, V Kartik, Abu Sebastian, and Angeliki Pantazi. High-speed multiresolution scanning probe microscopy based on lissajous scan trajectories. *Nanotechnology*, 23(18):185501, 2012.

[15] Juan P Wachs, Mathias Kölsch, and Deborah Goshorn. Human posture recognition for intelligent vehicles. *Journal of real-time image processing*, 5(4):231–244, 2010.

[16] Oliver Wulf and Bernardo Wagner. Fast 3d scanning methods for laser measurement systems. In *International conference on control systems and computer science (CSCS14)*, pages 2–5, 2003.

[17] S. Yohanan and K. E. MacLean. The role of affective touch in human-robot interaction: Human intent and expectations in touching the haptic creature. *International Journal of Social Robotics*, 4(2):163–180, 2012.

[18] Huijing Zhao and Ryosuke Shibasaki. A novel system for tracking pedestrians using multiple single-row laser-range scanners. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 35(2):283–291, 2005.

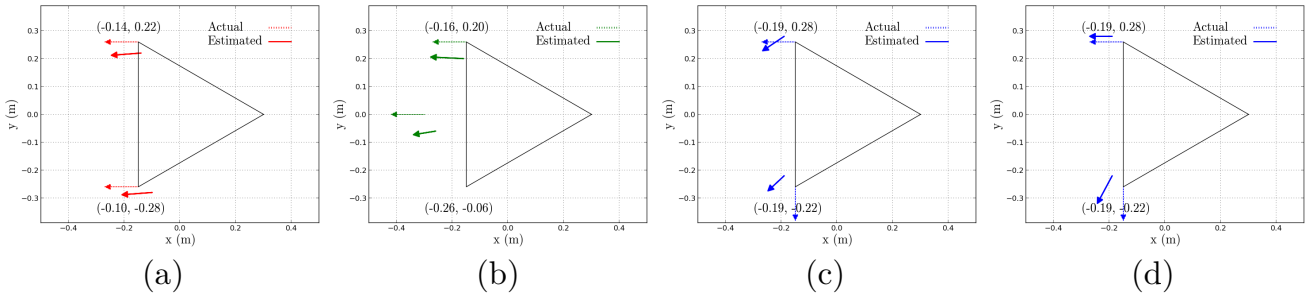

Fig. 6. **Estimated Multicontact Positions and Forces** are shown in the graphs. The forces in Fig (c) and (d) are estimated from the same data, but (d) uses a prior information of the left hand contact (upper one).

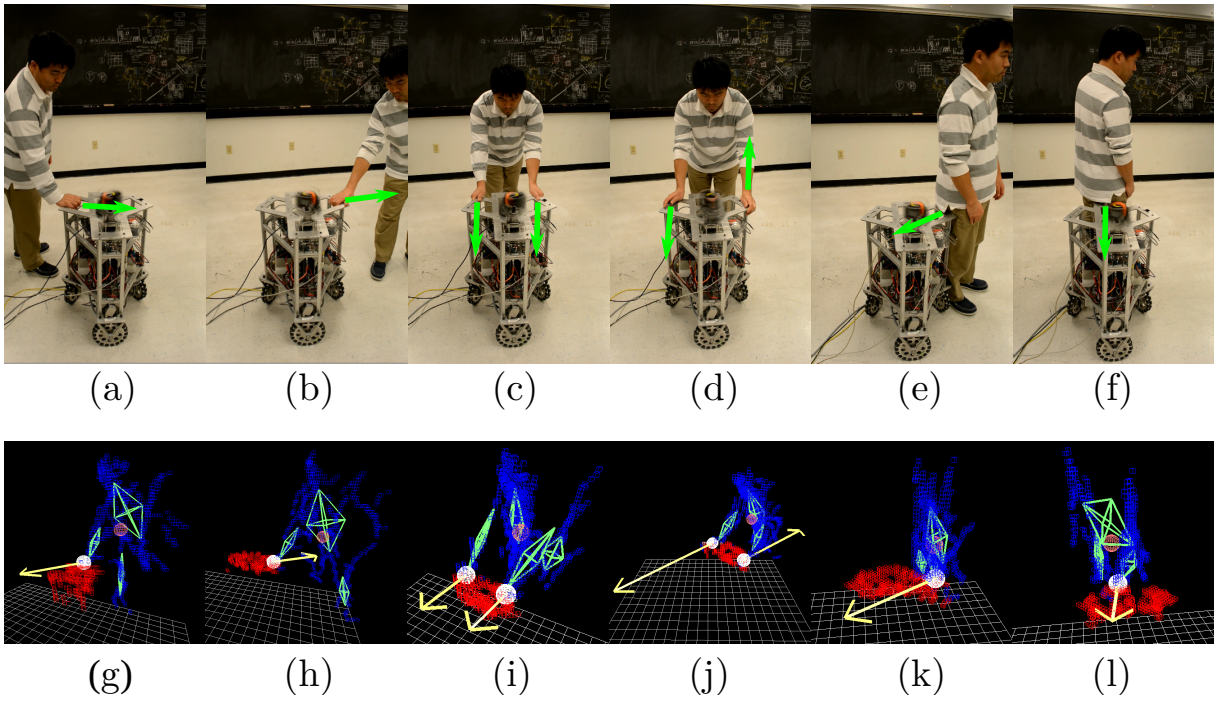

Fig. 7. **Contact Gesture Recognition** for multiple contact situations are shown in Fig (a)∼(f). Their corresponding occupancy maps are shown in Fig (g)∼(l), respectively. The white spheres show the contact locations, the green octahedrons represent the identified human body parts, and the yellow arrows are the estimated contact forces.

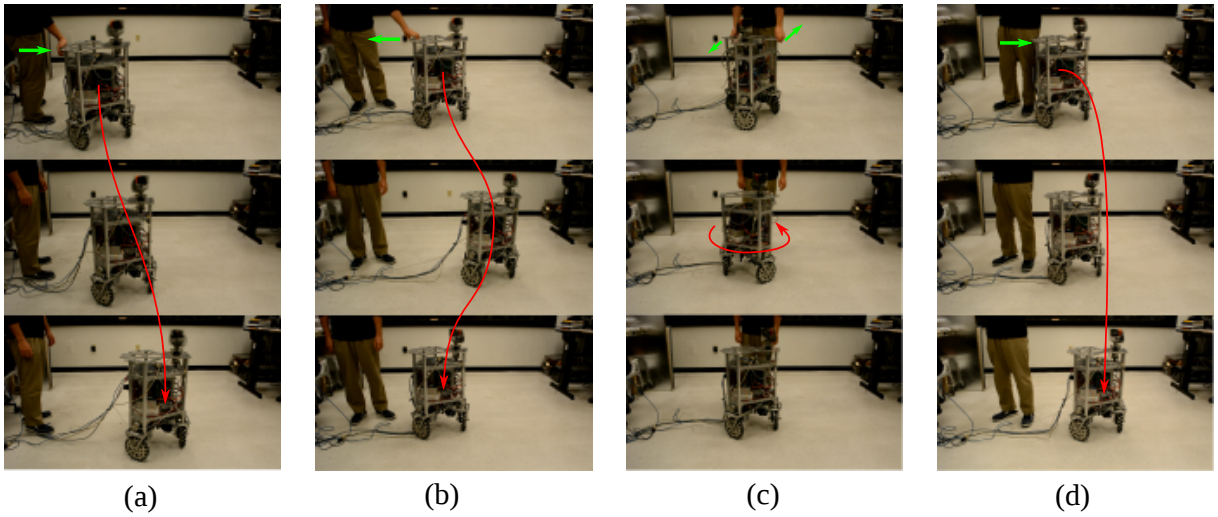

Fig. 8. **pHRI Experiments using Contact Gesture Recognition** demonstrate four use cases. Each use case starts from the top row. When the mobile platform detects contact forces, it operates following the predefined reaction table shown in Table I.