People tracking and re-identification by face recognition for RGB-D camera networks

Conference Paper · September 2017

DOI: 10.1109/ECMR.2017.8098689

5 authors, including:

Kenji Koide
Toyohashi University of Technology
8 PUBLICATIONS **6** CITATIONS
SEE PROFILE

Marco Carraro
University of Padova
8 PUBLICATIONS **7** CITATIONS
SEE PROFILE

Matteo Munaro
University of Padova
41 PUBLICATIONS **396** CITATIONS
SEE PROFILE

Jun Miura
Toyohashi University of Technology
313 PUBLICATIONS **2,349** CITATIONS
SEE PROFILE

People Tracking and Re-Identification by Face Recognition for RGB-D Camera Networks

Kenji Koide*[1], Emanuele Menegatti[2], Marco Carraro[2], Matteo Munaro[2], and Jun Miura[1]

***Abstract*— This paper describes a face recognition-based people tracking and re-identification system for RGB-D camera networks. The system tracks people and learns their faces online to keep track of their identities even if they move out from the camera's field of view once. For robust people re-identification, the system exploits the combination of a deep neural network-based face representation and a Bayesian inference-based face classification method. The system also provides a predefined people identification capability: it associates the online learned faces with predefined people face images and names to know the people's whereabouts, thus, allowing a rich human-system interaction. Through experiments, we validate the re-identification and the predefined people identification capabilities of the system and show an example of the integration of the system with a mobile robot. The overall system is built as a Robot Operating System (ROS) module. As a result, it simplifies the integration with the many existing robotic systems and algorithms which use such middleware. The code of this work has been released as open-source in order to provide a baseline for the future publications in this field.**

***Index Terms*— RGB-D Camera Network, People Tracking, Person Identification, Face Recognition.**

I. Introduction and Related Work

Camera network-based people tracking systems have attracted much attention in the computer vision community. They have been applied to various tasks, such as surveillance and human-robot interaction. One of the essential problems for people tracking is person re-identification. To keep track of people who left the camera view, systems have to re-identifies them when they re-appear in the view. In addition to that, a capability of identifying people over days (Long-term re-identification) is required in long-term service scenarios.

Many works proposed person re-identification methods for camera networks [1], [2], [3]. By combining appearance features which are robust to pose, illumination, and camera characteristics changes and sophisticated feature comparison methods, they achieved good people identification performance over multiple cameras. However, the proposed methods rely on the RGB information of each subject, thus, on the appearance of their clothes. Therefore, it is not possible to identify people over days and not applicable to long-term service scenarios.

Several soft biometrical features, such as gait [4], [5] and skeletal lengths [6], have been proposed for overcoming this problem. Since such biometrical features are specific to individuals and invariant over time, they enable to identify people over days. However, those features may be indiscriminative when several people have similar physiques. To reliably identify people, features for person re-identification have to be discriminative and robust to viewpoint changes. One of the most discriminative and reliable features to identify people is the face [7]. However, face features had not been widely used for person re-identification [2] since faces are not always visible, and we need to deal with pose, illumination, and expression (PIE) changes to apply it to camera networks. Some works used face features for person re-identification [8]. However, the use of face features was limited to assist appearance features, and those methods intended to be applied to short-term re-identification tasks.

It was difficult to solve the PIE issues by using traditional face features, such as EigenFace [9], Local Binary Patterns [10], and Scale-Invariant Features Transform [11]. On the other hand, recent deep neural network-based face representations [12], [13], [14] provide robust and discriminative face features and allow to reliably identify a person in presence of those issues. Although face visibility is still a hard problem, recent inexpensive consumer cameras allow to have a dense camera network. A person's face might be visible to any of the cameras under such settings.

In this paper, we propose a people tracking and re-identification system for RGB-D camera networks. The proposed system tracks people by using *OpenPTrack* [15], an RGB-D camera network-based people tracking framework, and learns their faces online to re-identify people who left the camera view once. For real-time performance, we employ a distributed people detection and face feature extraction structure. A PC is connected to each distributed RGB-D camera, and it detects people, extracts face features, and sends those data to a master PC. The master PC aggregates the information to track and re-identify people. This system also provides a predefined people identification capability: a set of people names and face images is given to the system, and it makes the association between tracks and the predefined people to know the names of the tracked people.

The contributions of this paper are two-fold. First, we propose a Bayesian inference-based face classification method. It allows to reliably classify a face according to the confidence of deep neural network-based face comparison results. Secondly, this work is an open source ROS [16] platform-based project [1], and it can easily be integrated with robotic

[1]Kenji Koide and Jun Miura are with the Department of Computer Science and Engineering at the Toyohashi University of Technology, Japan

[2]Emanuele Menegatti, Marco Carraro, and Matteo Munaro are with the Department of Information Engineering at the University of Padua, Italy

*indicates the corresponding author, email{koide@aisl.cs.tut.ac.jp}

A supplementary video is available at `https://goo.gl/yGRdEZ`

[1]The code is available at `https://goo.gl/ypILr7`

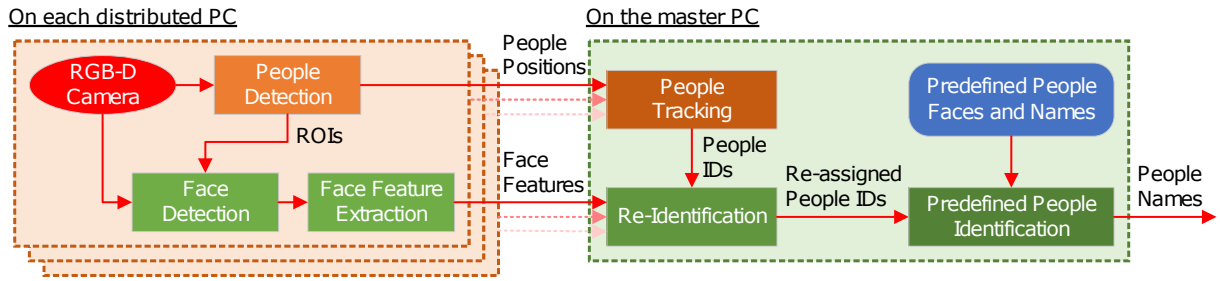


Fig. 1. System overview.

systems. We show the possibility of the integration of the system and robotic systems through the experiment.

The remainder of this paper is organized as follows. Sec. II describes an overview of the proposed method. Secs. III describes the proposed face recognition framework. Sec. IV describes the integration of the proposed system and a mobile robot system. Sec. V presents the evaluation and the experiment of the proposed system. Sec. VI concludes the paper and discusses future work.

II. System Overview

Fig. 1 shows an overview of the proposed system. The proposed system tracks people by using *OpenPTrack* [17], an RGB-D image-based people tracking framework, and re-identify people by face recognition based on *OpenFace* [18], a deep convolutional neural network-based face representation.

On each distributed PC connected to an RGB-D camera, we first perform RGB-D image-based people detection [19]. Then, we calculate ROIs from the detected people positions and detect faces on the ROIs. Face features are extracted from the detected face regions by using a deep neural network provided by *OpenFace*. In this system, the distributed PCs do not send raw camera images, but send only the detected people positions and extracted face features to the master PC. In this way, since those data is very light, we allow our system to be efficient and scalable to a large and dense camera network without bandwidth problems.

While the *OpenPTrack* master node tracks people from the detection positions it receives, our proposed system re-identifies people by face recognition. The re-identification part takes advantage of the tracks, which connect several frames from the same (still unknown) person and the face features associated to those frames. By integrating this information, the system can build online a descriptor for each person face. Then, it associates the tracks of the people with the learned faces. By referring the ID of the face associated with the track of a person, we can keep tracking the identity of the person even if he/she leaves and re-enters the camera view.

This system also provides a predefined people identification capability. A set of people names and faces are given to the system in advance, and the system associates the predefined people faces with the online learned faces. This capability allows to know a specific person's whereabouts and have a rich human-system interaction for applications like housework robots [20].

The proposed system is built on top of the Robot Operating System (ROS), thus, it can be easily combined with other robotic systems.

III. Proposed Method

A. People Detection and Tracking

We integrated the face recognition-based re-identification algorithm with *OpenPTrack* [15], a people tracking framework for RGB-D camera networks. This framework first detects people from point clouds obtained from RGB-D cameras placed in an environment. It removes the ground plane from a point cloud and then applies euclidean clustering to extract candidate clusters of human. Then, it judges whether a cluster is human or not by using an image-based SVM classifier on HOG features. This detection process is performed on each PC connected to an RGB-D camera, and then the detection results are aggregated on a master PC. The detected people are tracked by the combination of global nearest neighbor data association and Kalman filtering with a constant velocity motion model.

Fig. 2 shows a snapshot of *OpenPTrack*. As long as a person is visible from any of the RGB-D cameras, it provides reliable people tracking results. However, once a person leaves the field of view of all cameras, the system will assign a new track ID to the person whenever he/she will re-appear. In such cases, a person re-identification capability is necessary to recover the track of the person in such case.

B. Face Detection

To detect people faces in real-time, we take a top-down face detection approach. We first calculate ROIs from the people detection results provided by *OpenPTrack*. We project the rectangle with a fixed metric dimension (e.g., 0.2 m) to a detected human cluster's top position in the image. A HOG and cascaded SVM-based face detector [21] runs on the ROIs, and face features are extracted from the detected face regions. Fig. 3 shows an example of the face detection results. The green and red boxes indicate the ROIs calculated from the top positions of the detected people clusters, and the detected face regions, respectively.

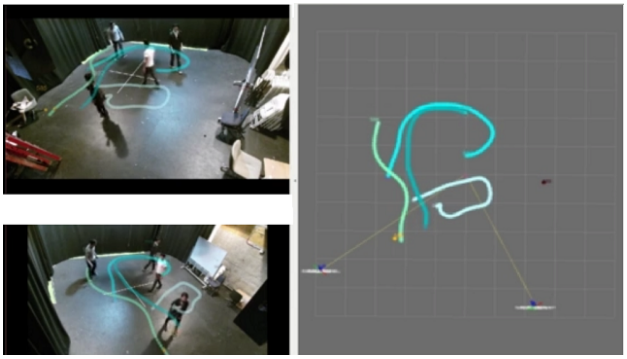


Fig. 2. OpenPTrack, a people tracking framework using a RGB-D camera network, provides robust people tracking. However, person re-identification capability is necessary to recover the track of a person left from the camera view. [17]

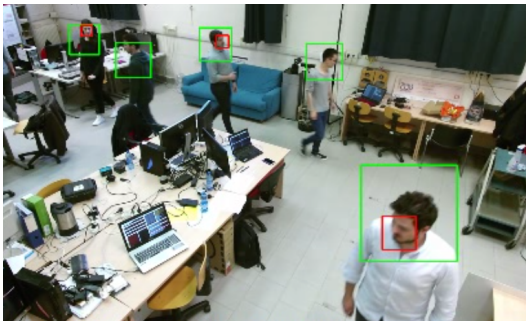


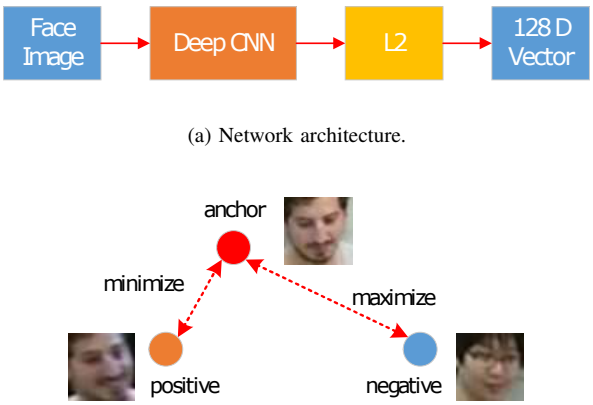
Fig. 3. An example face detection result. The green boxes indicate the ROIs calculated from the detected people positions. The red boxes indicate the detected face regions. Some faces are not detected due to their poses. However, by integrating the face detection results of multiple cameras, most faces will be observable from any of the cameras.

C. Deep Neural Network-based Face Features

We utilize *OpenFace* [18], a face recognition framework, to extract face features. The framework provides an implementation of *FaceNet* [13], a state-of-the-art deep convolutional neural network for face feature extraction. Fig. 4(a) shows the network architecture of *FaceNet*. The network first transforms a face image to a 128-dimensional feature vector by applying a deep convolutional neural network. Then, the feature vector is normalized so that its L2 norm becomes 1. The network is trained to minimize the triplet loss function [13]. This function takes three training data (i.e., *triplet*), a training data to be an *anchor*, a positive data with the same identity as the anchor, and a negative data with a different identity. The triplet loss function minimizes the L2 distance between the anchor-positive pair and maximizes the distance between the anchor-negative pair. As a result, faces of the same person are embedded close together, and face features of different persons are placed far from each other. Thereby, we can judge whether two faces have the same identity or not by calculating the L2 distance between their face features.

D. People Re-identification by Face Recognition

The system learns people faces online and associates the tracks of people and the learned faces to re-identify people who left the camera view by means of Algorithm 1. *face_list* is the list of online learned faces, and *track_list* is the list of people tracks. We first classify face images observed from the track of a person into the online learned faces. If the observed face images are classified as one of the online learned faces with high confidence, we assign the face's ID to the track. If the track is classified as an unknown person, we consider that he/she is a newly appeared person and give a new face ID to the track. While the track is alive, the system keeps the assigned face ID and learned face images. If the track of a person is lost, the system removes the track from *track_list* and moves the assigned face ID and images of the track to *face_list* to make it assignable to new tracks.



(a) Network architecture.

(b) Triplet loss function.

Fig. 4. *FaceNet* framework [13]. This framework employs a deep convolutional neural network and the triplet loss function to obtain discriminative face representations.

Algorithm 1 Assign a face ID to the track of a person

```
face_list is a set of (face_ID, face_images)
track_list is a set of (track_ID, face_ID, face_images)
for all track in track_list do
  add observed face images to track.face_images
  if track.face_ID has not been assigned then
    result ← classify(track.face_images, face_list)
    if result.confidence < threshold then
      continue
    else if result is known person then
      track.face_ID ← result.face_ID
      add result.face_images to track.face_images
      remove result from face_list
    else
      track.face_ID ← new face_ID
    end if
  end if
  if track is not alive then
    remove track from track_list
    add (track.face_ID, track.face_images) to face_list
  end if
end for
```

E. Bayesian Inference-based Face Classification

The simplest way to classify an observed face into registered faces is thresholding: if the distance between the observed face and a registered face is less than a threshold, the observed face is classified as the registered one. If several faces have the distances less than the threshold, the observed face is classified as the face with the minimum distance. Usually, this classification is performed for a certain number of observations (e.g., 5 face images) and the decision is made by majority voting. However, this method ignores the difference of the distances, and it may affect the classification accuracy when several faces have similar distances. The classification method should take into account the difference of the distances as the confidence of the classification for robust classification results.

To reliably classify observed faces, we propose a Bayesian inference-based face classification. Let $p(x_{ij})$ be the probability that the i-th track has the same identity as the j-th learned face, $p(x_{i0})$ be the probability that the i-th track's face has not been registered to the face list (the person is a new person), and $p(x_{0j})$ be the probability that the person who has the j-th registered face is not tracked (the person does not exist in the camera view). From this definition, we obtain the following constraints.

$$\sum_{k=0}^{N} p(x_{kj}) = 1 \quad (j \neq 0) \tag{1}$$

$$\sum_{k=0}^{M} p(x_{ik}) = 1 \quad (i \neq 0) \tag{2}$$

The probabilities can be represented as a table (see Fig. 5). We update this probability table with face images observed from people tracks. According to Bayesian theorem, the posterior probability of $p(x_{ij})$ under an observation y is given as:

$$p(x_{ij}|y) \propto p(x_{ij})p(y|x_{ij}) \tag{3}$$

We calculate the distances between the observed face image and the registered faces as follows and consider the distances as observations.

$$d_{ij} = \min_k \|F(x_i^t) - F(x_j^k)\|, \tag{4}$$

where, x_i^t is the face image observed from the tracker i at time t, x_j^k is the k-th face image of the j-th learned face, and $F(x)$ is the feature extraction function. To model the likelihood function $p(y|x_{ij})$, we randomly sampled correct and wrong face pairs from the LFW face database [22]. The number of the correct and wrong pairs are about 30000 and 60000, respectively. Fig. 6 shows the distributions of the L2 distances between the features extracted from the face pairs. We fit a skew normal distribution to each distribution by using maximum likelihood estimation and gradient decent method. The solid lines in Fig. 6 indicate the fitted skew normal distributions. We denote by $N_p(x)$ and $N_n(x)$ the skew normal distributions of the correct and wrong pairs, respectively. We calculate the likelihood $p(y|x_{ij})$ from a distance d_{ij} as:

$$p(y|x_{ij}) = \begin{cases} N_p(d_{ij}) & (j = i) \\ c \cdot N_n(d_{ij}) & (j = 0) \\ N_n(d_{ij}) & otherwise \end{cases} \tag{5}$$

where, c is a constant which models the tendency that an observed face image is produced from an unknown person. If we take a large c, the system tends to classify an observed face as an unknown face. In this work, we just use $c = 1$, however, it works well for most cases.

After updating the posterior probabilities, we normalize the probabilities so that they satisfy the equation (1) and (2) by using Shinkhorn iteration [23]. It first normalizes every row and then normalizes every column so that they sum up to 1. This normalization is repeated until the probabilities converge. After the normalization, if $p(x_{ij})$ is larger than a threshold (e.g., 0.95), we classify the i-th track as the j-th learned face.

F. Predefined People Identification

To allow a rich human-system interaction, the proposed system provides a predefined people identification capability.

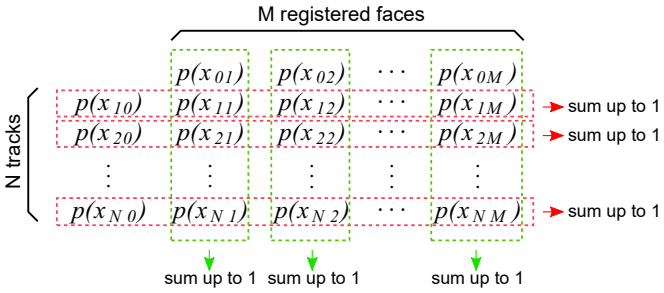


Fig. 5. The posterior probability table. The element at (i, j) is the posterior probability that i-th track has the same identity as the j-th learned face. Rows and columns are iteratively normalized so that they sum up to 1 by using Shinkhorn iteration.

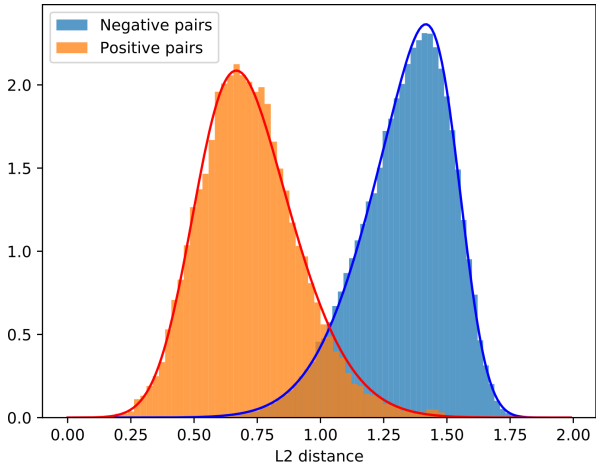


Fig. 6. The distributions of L2 distance of the positive and the negative face pairs. The solid lines indicate the skew normal distributions fitted to the distributions.

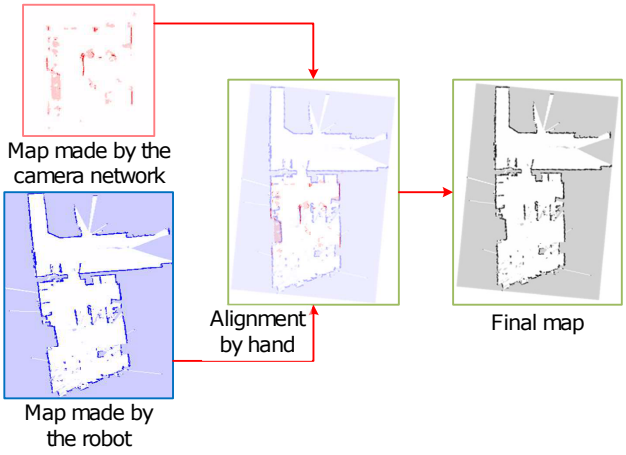


Fig. 7. Aligning an environmental map made by the robot to the one made by the camera network. An occupancy grid map is created from point clouds obtained from the camera network, and then the maps are manually aligned.

Predefined people data are given to the system as a set of people names and face images, and the system associates online learned faces and the predefined people.

We calculate the distance between face images of a predefined person and learned face images, and if the distance is smaller than a threshold, the system associates the predefined person and the learned face. The distance is calculated as the minimum distance of the distances between all combinations of the predefined person's face images and the learned face images.

IV. Cooperation with a Mobile Robot

Since the proposed system has been implemented as ROS modules, it can easily cooperate with robotic systems. In this work, we integrate the proposed system with a mobile robot to demonstrate a human-system interaction service. We use a Turtlebot2 robot equipped with a laser range finder (see Fig. 11). By using ROS navigation stack [24], we made the robot create an environmental map and estimate its current pose. To operate the robot with the camera network, it is necessary to know the transformation between the created map and the camera network coordinates. We first extract points in a certain height range (e.g., 0.1 - 0.3m) from the camera network and create an occupancy grid map. Then, we manually align the map made by the robot to the one made by the camera network (see Fig.7). Note that this map alignment can be semi-automated by giving an initial guess by hand and applying ICP matching between the maps. Fig. 8 shows a snapshot of the integrated system. We can see the robot, the point clouds obtained by the Kinects, and the tracked people in the same view.

V. Experiments

A. Person Re-identification Experiment

We conducted long-term person re-identification experiments. Two RGB-D image sequences were recorded at different days. The subjects changed their clothes (and hair styles) between the recordings, thus, appearance-based re-identification is not effective for this dataset. In each of the sequences, six subjects are walking in the environment. Four of them appear in both the sequences. Fig. 9 shows a snapshot of the experiments. We put three Kinects so that they cover the environment. Table I shows a summary of the dataset. The durations of the sequences are 162 and 218 [s]. The subjects often moved out from the camera's field of view, and the system lost track of people 49 and 58 times in the sequence 1 and 2, respectively.

The proposed method is applied to the sequences. For comparison, we also applied the proposed method with

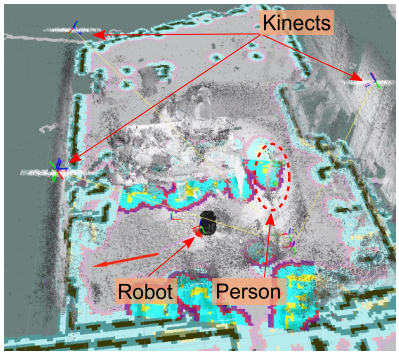


Fig. 8. The integration of the system and a mobile robot. We can see the robot, the point clouds obtained from the Kinects, and the tracked person in the same view.

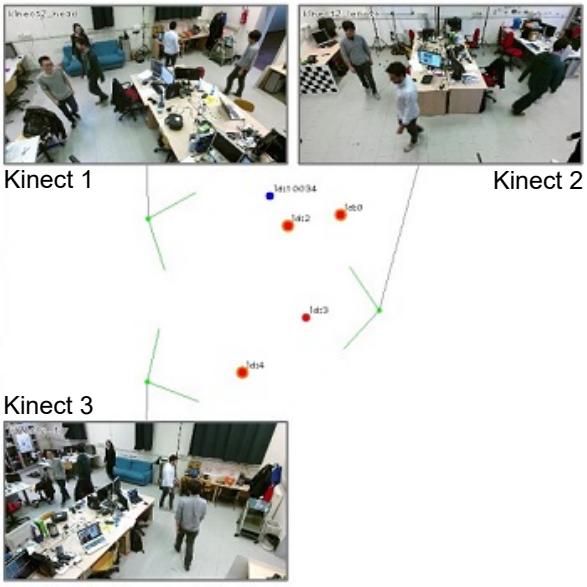


Fig. 9. A snapshot of the experiments. Three Kinects are placed so that they cover the environment. The dots indicate the tracked people.

TABLE I
Long-term re-identification dataset summary

	duration [s]	# of subjects	# of lost
Seq. 1	162	6	49
Seq. 2	218	6	58

thresholding instead of the Bayesian inference and a traditional face recognition method with a landmark detection and SIFT features [25]. In this experiment, the face detection and the feature extraction took about 15 msec per frame on each distributed PC, and the re-identification took about 10 msec on the master PC.

Table II shows a summary of the re-identification results. *success* and *failure* in Table II mean the number of tracks successfully and wrongly re-identified. Since *OpenFace* discards a face image when it couldn't detect the landmarks of the face, it yields fewer face features than [25], and it took much time to re-identify a person from when he/she appears than the SIFT-based method. However, the recognition accuracy of the proposed methods significantly outperforms [25]. While the limitation of the face view faces make the SIFT-based method fail to identify people, the proposed methods achieve the high identification accuracies thanks to the robustness of the deep neural network-based face representation to the face view change.

With the simple thresholding scheme, the system has to make a decision when it observes a certain number of face images. This affects the re-identification accuracy when several registered faces show similar distances to an observed face image. On the other hand, with the proposed Bayesian inference, the system can wait to make a decision until a significant difference of the faces is observed. In addition to that, once a significant difference is observed, the system can immediately classify the observed face. As a result, the proposed method with the Bayesian inference could improve both the recognition accuracy and the average re-identification time. Fig. 10 shows examples of the long-term re-identification results (i.e., re-identification between the sequence 1 and 2). The system successfully re-identified most of the subjects even if they changed their clothes (and hair styles). However, it wrongly identified a newly appeared person as an existing person due to their similar face appearances (beards and brows). It was the only failure case in this experiment. *recover rate* in Table II shows the rate of the successfully re-identified tracks among all the tracks. It shows a limitation of the face recognition-based re-identification: while it shows a good re-identification accuracy, it cannot re-identify a person when his/her face is not visible. One possible way to address this problem is combining the face recognition-based re-identification with other re-identification methods, such as appearance-based and soft biometric-based methods. It allows taking advantage of the high recognition accuracy of the face recognition and visibility independence of such methods.

TABLE II

EVALUATION OF RE-IDENTIFICATION ACCURACY

	success	failure	re-id time	recover rate
[25]	34	54	2.12	0.318
ours w/o Bayesian	51	7	4.59	0.477
ours w/ Bayesian	60	1	4.24	0.561

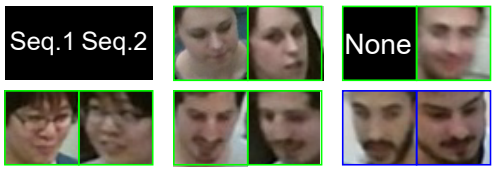


Fig. 10. Examples of the long-term re-identification results. The green boxes indicate that the person is correctly re-identified. The blue boxes indicate that the right side face is wrongly identified as the left side face.

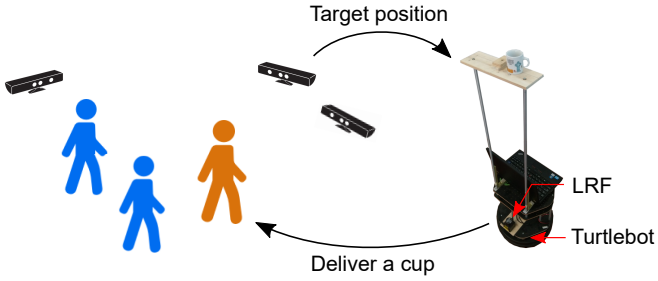


Fig. 11. The cup delivery experiment. The system tells the target person position to the robot, and the robot delivers a cup to him.

B. Experiment with a Mobile Robot

To show the possibility of the integration of the proposed system and a mobile robot, we conducted an experiment with a daily service robot scenario. The task of the robot is to deliver a cup to a specific target person. A pair of the target person's name and a face image is given to the system in advance. The system identifies the target person among the others and tells his/her position to the robot. Then, the robot moves toward the target person's positions to deliver the cup to him/her (see Fig. 11).

Fig. 12 shows the experimental setting. The number of the subjects is three, and each subject stands in front of a Kinect and stares at it. We give a list of people names and face images of the subjects and other three people. Thus, the number of the predefined people is six. We conducted the experiment three times while changing the target.

Fig. 13 shows a snapshot of the experiment. While the target person stared at the Kinect, the system successfully identified him as the target person (Fig. 13 (a)). Then, the system told his position and name to the robot, and the robot moved toward him (Fig. 13 (b)). After arriving at his position, the robot called his name and told him to take the cup (Fig. 13 (b)), and the target person took the cup on the robot (Fig. 13 (d)).

In all the experiments, the system successfully identified the target person, and the robot delivered the cup to the target person. The experimental result shows that the proposed system can be integrated with a mobile robot system, and it allows to have a rich human-system interaction.

VI. CONCLUSIONS AND FUTURE WORK

This paper has described a face recognition-based people tracking and re-identification system for RGB-D camera networks. The proposed system utilizes *OpenPTrack* and *OpenFace* to track and recognize their faces. We proposed

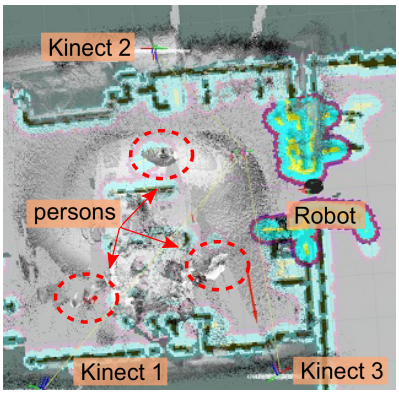
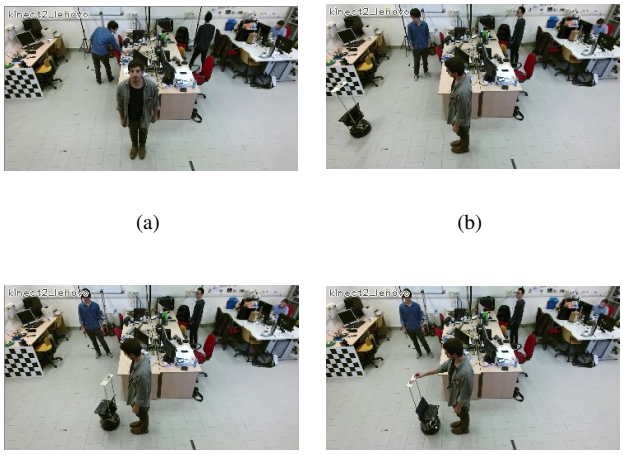


Fig. 12. The setting of the experiment with a mobile robot. A person stands in front of each Kinect and stare at it. The system identifies a target person and tells his position to the robot, and then the robot moves toward him to deliver a cup.



(a) (b)

(c) (d)

Fig. 13. A snapshot of the cup delivery experiment. The target person is identified by the camera network, and then his position is told to the robot. The robot moves toward the target person, and after arriving, the robot calls his name and tells him to take the cup on the robot.

a Bayesian inference-based face classification method for reliable re-identification. We evaluated the re-identification performance of the proposed system and conducted an experiment to show the possibility of the integration of the proposed system with robotic systems.

Face visibility is still a hard problem. The system cannot identify a person if the person hides his/her face intentionally. To deal with such situations, we are planning to combine the face features with features which do not suffer from the visibility problem, such as appearance and skeletal features.

ACKNOWLEDGEMENT

This research was partially supported by Omitech srl under the O-robot grant and the Leading Graduate School Program R03 of MEXT.

REFERENCES

[1] R. Mazzon, S. F. Tahir, and A. Cavallaro, "Person re-identification in crowd," *Pattern Recognition Letters*, vol. 33, no. 14, pp. 1828–1837, oct 2012.

[2] A. Bedagkar-Gala and S. K. Shah, "A survey of approaches and trends in person re-identification," *Image and Vision Computing*, vol. 32, no. 4, pp. 270–286, apr 2014.

[3] L. Nanni, M. Munaro, S. Ghidoni, E. Menegatti, and S. Brahnam, "Ensemble of different approaches for a reliable person re-identification system," *Applied Computing and Informatics*, vol. 12, no. 2, pp. 142–153, jul 2016.

[4] A. Bedagkar-Gala and S. K. Shah, "Gait-assisted person re-identification in wide area surveillance," *Computer Vision - ACCV 2014 Workshops*, pp. 633–649, 2015.

[5] K. Koide and J. Miura, "Identification of a specific person using color, height, and gait features for a person following robot," *Robotics and Autonomous Systems*, vol. 84, pp. 76–87, oct 2016.

[6] M. Munaro, A. Fossati, A. Basso, E. Menegatti, and L. V. Gool, "One-shot person re-identification with a consumer depth camera," in *Person Re-Identification*. Springer, 2014, pp. 161–181.

[7] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Computing Surveys*, vol. 35, no. 4, pp. 399–458, dec 2003.

[8] A. Bedagkar-Gala and S. K. Shah, "Part-based spatio-temporal model for multi-person re-identification," *Pattern Recognition Letters*, vol. 33, no. 14, pp. 1908–1915, oct 2012.

[9] M. Turk and A. Pentland, "Face recognition using eigenfaces," in *Conference on Computer Vision and Pattern Recognition*. IEEE, 1992, pp. 71–86.

[10] G. Zhang, X. Huang, S. Z. Li, Y. Wang, and X. Wu, "Boosting local binary pattern (LBP)-based face recognition," in *Advances in Biometric Person Authentication*. Springer, 2004, pp. 179–186.

[11] C. Geng and X. Jiang, "Face recognition using SIFT features," in *International Conference on Image Processing*. IEEE, 2009, pp. 3277–3280.

[12] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 Classes," in *Conference on Computer Vision and Pattern Recognition*. IEEE, 2014, pp. 1891–1898.

[13] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Conference on Computer Vision and Pattern Recognition*. IEEE, 2015, pp. 815–823.

[14] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *British Machine Vision Conference*. BMVA, 2015, pp. 41.1–41.12.

[15] M. Munaro, F. Basso, and E. Menegatti, "OpenPTrack: Open source multi-camera calibration and people tracking for RGB-d camera networks," *Robotics and Autonomous Systems*, vol. 75, pp. 525–538, jan 2016.

[16] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. IEEE, 2009, p. 5.

[17] "Openptrack — towards collective computing." [Online]. Available: http://openptrack.org/

[18] A. Brandon, B. Ludwiczuk, and S. Mahadev, "Openface: A general-purpose face recognition library with mobile applications," CMU-CS-16-118, CMU School of Computer Science, Tech. Rep., 2016.

[19] M. Munaro and E. Menegatti, "Fast RGB-d people tracking for service robots," *Autonomous Robots*, vol. 37, no. 3, pp. 227–242, 2014.

[20] M. Carraro, M. Antonello, L. Tonin, and E. Menegatti, "An open source robotic platform for ambient assisted living," in *Italian Workshop on Artificial Intelligence and Robotics*, 2015, pp. 3–18.

[21] "Dlib c++ library." [Online]. Available: http://dlib.net/

[22] G. B. H. E. Learned-Miller, "Labeled faces in the wild: Updates and new reporting procedures," University of Massachusetts, Amherst, Tech. Rep. UM-CS-2014-003, May 2014.

[23] T. Tamaki, M. Abe, B. Raytchev, and K. Kaneda, "Softassign and EM-ICP on GPU," in *International Conference on Networking and Computing*. IEEE, nov 2010.

[24] "ROS navigation stack." [Online]. Available: http://wiki.ros.org/navigation

[25] G. Pitteri, M. Munaro, and E. Menegatti, "Depth-based frontal view generation for pose invariant face recognition with consumer RGB-D sensors," in *Intelligent Autonomous Systems Conference*. Springer, 2016.