IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 23, 2021, accepted June 3, 2021, date of publication June 17, 2021, date of current version June 28, 2021.

*Digital Object Identifier 10.1109/ACCESS.2021.3090170*

# Edge-Based Meta-ICP Algorithm for Reliable Camera Pose Estimation

**CHUN-WEI CHEN[1], JONAS WANG[2], AND MING-DER SHIEH[1], (Member, IEEE)**
[1]Department of Electrical Engineering, National Cheng Kung University, Tainan 70101, Taiwan
[2]Himax Technologies, Inc., Tainan 74148, Taiwan

Corresponding author: Ming-Der Shieh (shiehm@mail.ncku.edu.tw)

**ABSTRACT** Camera pose estimation is crucial for 3D surface reconstruction and augmented reality applications. For systems equipped with RGB-D sensors, the corresponding transformation between frames can be effectively estimated using the iterative closest point (ICP) algorithms. Edge points, which cover most of the geometric structures in a frame, are good candidates for control points in ICP. However, the depth of object contour points is hard to accurately measure using commercial RGB-D sensors. Inspired by the model-agnostic meta-learning (MAML) algorithm, this work proposes a meta-ICP algorithm to jointly estimate the optimal transformation for multiple tasks, which are constructed by sampled datapoints. To increase task sampling efficiency, an edge-based task set partition algorithm is introduced for constructing complementary task sets. Moreover, to prevent ICP from being trapped in local minima, a dynamic model adaptation scheme is adopted to disturb the trapped tasks. Experimental results reveal that the probability of unstable estimations can be effectively reduced, indicating a much narrower error distribution of repeated experiments when adopting re-sampled points. With the proposed scheme, the overall absolute trajectory error can be improved by more than 30% as compared to the related edge-based methods using frame-to-frame pose estimation.

**INDEX TERMS** Camera pose estimation, iterative closest point, model-agnostic meta-learning.

## I. INTRODUCTION

In augmented reality (AR) applications, the quality of virtual content registration highly relies on the understanding of the camera viewing angle and position. The camera trajectory, consisting of a series of camera poses, needs to be reliably estimated for high-quality virtual content rendering. By estimating the transformation between the incoming frame and an estimated frame, the corresponding camera pose of the incoming frame can then be obtained. Commercialized light-weight RGB-D sensors are commonly equipped in nowadays AR systems for accessing the point clouds of the surrounding environment. By using the point clouds, transformation between frames can be efficiently estimated using point cloud registration methods such as the iterative closest point (ICP) algorithm [1], [3].

ICP computes the relative transformation between two point clouds by iteratively minimizing the distance metric between correspondences estimated according to the spatial distance. It tends to become trapped in local minima due to the adopted nonlinear local search strategy [2]. Adopting the point-to-plane error metric [3] instead of the point-to-point metric [1] makes ICP less likely to fall into local minima, and can take advantage of convergence speed [4]. In point-to-plane error metric, an error vector is projected onto the corresponding normal vector; thus correspondences located at smooth regions do not affect the error metric. The surface can thus freely slide away from the trapped location. However, when the geometry constraint of the correspondences is insufficient, ICP might not be able to converge stably. To improve stability, correspondences are sampled [5] or weighted [6] by analyzing the covariance matrix. To guarantee retaining a sufficient number of constraining points after correspondence pairing and rejection, Gelfand [2] proposed a sampling strategy for the input mesh. However, constraint-based sampling strategies might over-emphasize noisy regions while trying to improve the geometric constraint.

Edge points preserve most of the structure details in a scene [7]. Assuming the edge points are consistently detected, an optimized transformation can then be obtained by

The associate editor coordinating the review of this manuscript and approving it for publication was Yizhang Jiang.

minimizing the distance to the closest edge in the target image. Tarrio [10] presented an efficient matching algorithm that searches for the corresponding edge point along the normal direction. The matching procedure can further be simplified by pre-computing the edge distance map for the target image using the distance transform [11]. Therefore, the distance to the closest edge point can be efficiently obtained by transforming the source edge points onto the target edge distance map [12]. Recently, Schenk [13] analyzed the influence of adopting different machine-learned edges, such as the structured edges (SE) [14] or a CNN-based edge detector [15]. Their experimental results revealed that machine-learning-based edge detectors can provide higher repeatability for detecting edges as compared to the traditional Canny edge detector [16]. However, even though promising results are obtained using machine-learning-based edge detectors, the Canny edge detector still outperforms them in some cases.

Recently, meta-learning approaches have gained increasing attention due to the feasibility of tackling few-shot learning problems. For example, Finn [18] introduced an efficient model-agnostic meta-learning algorithm (MAML) that is applicable to general gradient-updated learning problems including classification, regression, and reinforcement learning. To prevent overfitting in learning from few samples, MAML seeks for initial model parameters which are suitable for fast adapting to the desired task through few gradient updates. By jointly training a representative model from the prepared training tasks, the meta-trained model can quickly adapt to unseen tasks in the meta-testing phase. Based on MAML, lots of extension works have been presented in the literature such as the first-order variant [20], the probabilistic extension [19], the multimodal extension [23], and the approach using unsupervised task construction [24]. To realize the root cause of MAML effectiveness, Raghu [26] analyzed the MAML-trained model and concluded that the model effectiveness is primarily due to the feature reusability rather than the rapid learning ability.

Instead of adopting CNN-based edge detectors as presented in [15] to learn the edges, this work explores how to efficiently apply the concept of meta-learning strategy [18] to the edge-based ICP algorithm for improving the accuracy and reliability of pose estimation. Conventional low-complexity edge detectors were adopted in this work to demonstrate the effectiveness of introducing the meta-learning strategy. To the best of our knowledge, this is the first work to investigate the combination of these two schemes. Moreover, since the meta-learned model can be trained more effectively by using properly selected training tasks [27], this work also presents schemes to construct comprehensive data sets that are employed to sample datapoints for training tasks. For example, due to the limitation of commercialized light-weight depth sensors, the depth of boundary points might be unstable, and a straight edge might appear as a zig-zag line in a depth image [9]. Thus, we extended the extracted depth edges by considering points in regions near the edges to obtain a more reliable estimation while trying to retain the abundant structures provided by edge points.

This paper presents an edge-based meta-ICP algorithm to jointly learn the transformation from multiple edge types. The main contributions of this work are summarized as follows:

1. This work presents a novel ICP algorithm based on the meta-learning strategy. An effective edge-based task set partition algorithm is introduced to construct complementary tasks for meta-training. Moreover, an entropy-based objective function is introduced to balance the size of task sets.
2. To prevent ICP from being trapped in local minima, this work introduces a dynamic model adaptation scheme to disturb the trapped tasks by considering the parameters of the other tasks.
3. Using the proposed schemes, experimental results show that the worst-case performance of the absolute trajectory error, evaluated by repeated experiments, can be effectively suppressed. For frame-to-frame pose estimation, the overall absolute trajectory error is improved by more than 30% in comparison with the results derived by REVO [13], Canny VO [25], and ORB2 VO [22].

The rest of this paper is organized as follows: Section II briefly reviews the depth edge detection, the ICP algorithm, and the MAML algorithm. Section III presents the proposed edge-based meta-ICP algorithm. Section IV shows the experimental results and comparisons with related methods. Finally, Section V concludes this work.

## II. BACKGROUND

### A. DEPTH EDGE DETECTION

Depth edge points, which preserve most of the details in a scene structure, are good candidates used in geometric alignment. By evaluating the depth discontinuity between neighboring pixels, edge pairs can then be identified [7, 8]. Pixels $(i,j)$ belong to an edge pair only if pixel $i$ and the nearest valid depth pixel $j$ satisfy the following equation:

$$\left|z_i - z_j\right| > T_D \cdot \min\left(z_i, z_j\right), \tag{1}$$

where $T_D$ is a positive sensitivity constant and $z_k$ denotes the depth value of pixel $k$. Because the pixels in Kinect depth images are calculated from inverse disparity values, which are normalized and quantized to 11-bit integer values, the depth uncertainty caused by the disparity quantization error is not uniformly distributed over the depth values. Moreover, the space between two readable depth values is not constant but proportional to the true distance [28]. Thus, a proportional depth threshold is needed for determining the real depth discontinuity.

The edge with a smaller depth value is an occluding edge, and the other is an occluded edge. To effectively find the depth edges in a depth image, Bose [8] employed a row-column search strategy to find the depth discontinuity. When conducting a column search on row $v_i$, every pixel $i = (u_i, v_i)^{\mathrm{T}}$ that

has a valid depth value is compared with the most recently found valid depth pixel $j = (u_i - \Delta u, v_i)^{\mathrm{T}}$. A row search is conducted in the same manner.

### B. ICP ALGORITHM

The ICP algorithm is an iterative approach that can be applied to align two point clouds by repeatedly refining the relative rigid-body transform. The process in each iteration of the conventional ICP algorithm consists of the following steps.

#### 1) CORRESPONDENCE PAIRING

For the $i$-th selected point in the source cloud $P$, find the closest point $q_i$ as its correspondence in the destination cloud $Q$. The correspondence determination problem can be formulated as:

$$q_i = \underset{q \in Q}{\operatorname{argmin}} \left\| T^t p_i - q \right\|, \tag{2}$$

where $T^t = [r|\tau]$ is the initial transformation used for the $t$-th iteration. $r$ and $\tau$ are the rotation matrix and the translation matrix, respectively. The transformation matrix $T$ can then be decomposed into the transformation vector $\theta = [\theta_r, \theta_\tau]$, where $\theta_r$ and $\theta_\tau$ stand for the rotation angles around the xyz axes and the translation vector, respectively.

However, if the closest point is far away from the true correspondence, the transformation will be incorrectly estimated when using correspondences without removing the outlier pairs. Pulli [17] suggested using dynamic threshold $T_s$ and hard threshold $T_h$ to reject outliers. The hard threshold is applied to discard pairs with a distance larger than $T_h$ and the dynamic one is used to keep the $T_s$ percentage of the closest correspondences, aiming to remove those spurious pairs in the early iterations of ICP.

#### 2) TRANSFORMATION ESTIMATION

After removing the outliers, the remaining correspondence pairs are used to estimate the relative transformation, also denoted as the incremental transformation $T^*$. The $T^*$ of the $t$-th iteration derived from minimizing the predefined cost function such as the point-to-plane error metric [3] can then be calculated by

$$T^* = f_T\left(D, T^t\right) = \underset{T}{\operatorname{argmin}} \sum\nolimits_i \left\| \left(T \cdot T^t p_i - q_i\right) \cdot n_{q_i} \right\|, \tag{3}$$

where $n_{q_i}$ is the estimated normal vector of point $q_i$, and $D = \{p_i, q_i, \dots\}$ is the set of the correspondence pairs.

#### 3) TRANSFORMATION UPDATES

The derived incremental transformation is applied to transform the source frame and update the transformation as $T^{t+1} \leftarrow T^* T^t$.

The iteration process terminates when it reaches the maximum iteration number.

### C. MAML ALGORITHM

MAML algorithm, which consists of two optimization loops, aims at learning a representative model that can fast adapt to new tasks. In the inner loops, the adapted parameters for each sampled training task will be computed based on a shared model. Then, the shared model is trained by optimizing the performance across all sampled training tasks using the adapted model parameters in the outer loops. The MAML algorithm iteratively refines the shared model until the model is broadly applicable to most of the sampled training tasks within few gradient updates. The complete MAML algorithm for task adaption using single gradient update is described in Algorithm 1.

**Algorithm 1** MAML for Supervised Learning [17]

**Require:** $p(T)$: distribution over tasks
**Require:** $\alpha$, $\beta$: step size hyper-parameters
// the model is represented by a parameterized function $f_\theta$ with parameters $\theta$
// the cross-entropy loss for task $i$ using model $f_\theta$ with data $D$ is denoted as $L_i(f_\theta, D)$
Randomly initialize $\theta$
**While** not done **do**
  Sample batch of tasks $T_i \sim p(T)$
  **for all** $T_i$ **do**
    Sample $K$ datapoints $D_i = \{x_k, y_k\}$, $k = 1 \sim K$, from $T_i$
    Evaluate $\nabla_\theta L_i(f_\theta, D_i)$
    Compute adapted parameters using one gradient update:

$$\theta_i' = \theta - \alpha \cdot \nabla_\theta L_i(f_\theta, D_i)$$

    Sample another $K$ datapoints $D_i' = \{x_k, y_k\}$, $k = 1 \sim K$, from $T_i$
  **end for**
  Update $\theta \leftarrow \theta - \beta \cdot \nabla_\theta \sum L_i\left(f_{\theta_i'}, D_i'\right)$
**end while**

## III. PROPOSED EDGE-BASED META-ICP ALGORITHM

Since the depths of object boundaries are hard to be precisely measured using commercial depth sensors, this work leverages the concept of meta-learning techniques to improve the robustness of the ICP algorithm. Moreover, conventional low-complexity edge detectors were adopted in this work to demonstrate the effectiveness of introducing the meta-learning strategy.

MAML algorithm is a promising technique adopted for few-shot learning problems. By learning from the prepared training tasks, MAML algorithm explores an initial model that can be quickly adapted to new tasks. Inspired by MAML algorithm, this work proposes a novel edge-based meta-ICP algorithm to jointly learn the optimal transformation across multiple tasks. Through meta-training, the impact of individual spurious estimations caused by noisy depths can thus be suppressed. Since the meta-training process can provide a reliable transformation with respect to all the training tasks, ICP algorithm would be much robust to noisy depths

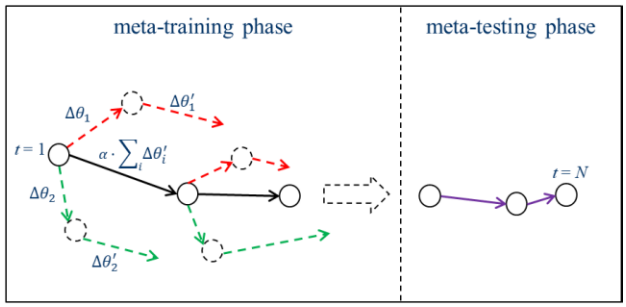


**FIGURE 1.** Paradigm of the proposed meta-ICP algorithm with two task sets configuration in which circles denote camera poses. All non-black arrows represent the transformation estimated using (3), while adopting data from different task sets are marked in different colors.

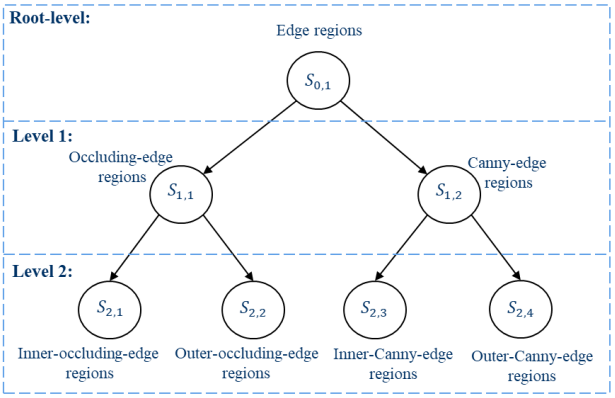


**FIGURE 2.** Hierarchical structure of the edge region partition in which $S_{i,j}$ denotes the $j$-th task set at the $i$-th hierarchical level.

when it is initialized using the meta-trained transformation. Moreover, to effectively sample useful tasks for performing meta-training, an edge-based task set partition algorithm is introduced. Data for each task are then sampled from the corresponding task set. A paradigm of the proposed edge-based meta-ICP algorithm is depicted in Fig. 1.

As shown in Fig. 1, the meta-ICP iterations are divided into the meta-training phase and the meta-testing phase. In the meta-training phase, transformation vectors are estimated according to the corresponding adapted tasks. An optimal transformation vector, denoted by the black arrow, is then estimated across all tasks. Once the meta-training is completed, the globally covered task set is used for performing ICP.

### A. EDGE-BASED TASK SET PARTITION ALGORITHM

Since the optimal transformation for each frame is different, the ground-truth corresponding point for the sampled source point cannot be prepared offline. Following the ICP assumption, the nearest point in the target cloud is selected as the corresponding point. A task is composed of a set of source points and their corresponding points, which can be calculated using (2). Since the output is mapped using the selected input point, the tasks are defined by the input source points discussed as follows.

By definition, any subset of the source cloud can be treated as a task. However, the distribution of the sampled tasks would be similar when directly adopting the random sampling strategy. To spread the risk of spurious estimation, this work explores to select tasks that are complementary to each other. By partitioning the source frame into regions with complementary features, complementary tasks can then be effectively constructed by sampling points according to the partitioned regions.

Based on the extracted edge information, this work introduces a two-level partitioning scheme to construct complementary task sets. At the first level, two edge types are considered: the occluding-edge regions and the Canny-edge regions. Each edge type is further partitioned into the inner-edge regions and the outer-edge regions at the second level. The hierarchical structure is depicted in Fig. 2.

In this work, the occluding-edge detector presented in [8] was employed to extract major object boundaries which have significant depth discontinuity as compared to neighboring pixels. Luminance edges with large gradients were extracted by using the Canny-edge detector [16]. Let the binary edge map of detected occluding-edge pixels and Canny-edge pixels be denoted as $G$ and $L$, respectively. Moreover, to extend edge features, the edge pixels are dilated to the edge regions which are further partitioned into two parts: inner regions and outer regions. The binary edge map of inner-occluding-edge regions $M_{2,1}$ and the inner-Canny-edge regions $M_{2,3}$ can be written as:

$$\begin{aligned} M_{2,1} &= G \oplus B_{I,G}, \\ M_{2,3} &= L \oplus B_{I,L}, \end{aligned} \tag{4}$$

where the symbol $\oplus$ stands for the morphological dilation operator. $B_{I,G}$ and $B_{I,L}$ are the inner dilation kernels for $G$ and $L$, respectively. The outer regions for $G$ and $L$ can be extracted using another dilation kernels as described in the following equations:

$$\begin{aligned} M_{2,2} &= \left(D \oplus B_{O,G}\right) \& \left(\sim M_{2,1}\right), \\ M_{2,4} &= \left(L \oplus B_{O,L}\right) \& \left(\sim M_{2,3}\right), \end{aligned} \tag{5}$$

where $B_{O,G}$ and $B_{O,L}$ are the outer dilation kernels for $G$ and $L$, respectively. Finally, the set $S_{2,j}$ can be constructed using the corresponding edge map $M_{2,j}$, for $1 \leq j \leq 4$. To avoid repeatedly sampling data in certain regions, the intersection of the sets within the same hierarchical level is discarded. Finally, the parent node is built by the union of all child nodes and the corresponding binary edge map can be written as:

$$S_{i,j} = S_{i+1,j} \cup S_{i+1,j+1}. \tag{6}$$

According to the structure of a captured scene, the amount of occluding edges and Canny edges might be quite imbalanced. Sampling points from task sets with insufficient points might also lead to unstable estimation. To balance the sizes of the partitioned task sets, the four dilation kernels are adjusted

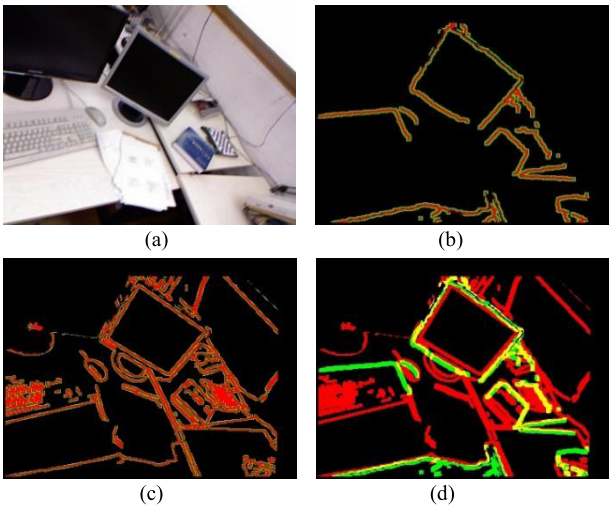


**FIGURE 3.** Results of edge region partition selected from the TUM RGB-D SLAM benchmark sequence (freiburg1_rpy) [21]. (a) Color image, (b) corresponding inner-occluding-edge regions $S_{2,1}$ (red) and outer-occluding-edge regions $S_{2,2}$ (green), (c) inner-Canny-edge regions $S_{2,3}$ (red) and outer-Canny-edge regions $S_{2,4}$ (green), and (d) root-level edge regions $S_{0,1}$ which contain occluding-edge regions $S_{1,1}$ (green), Canny-edge regions $S_{1,2}$ (red), and the intersection regions (yellow).

by solving the proposed entropy-based objective function defined as:

$$B^* = \underset{B}{\operatorname{argmin}} \sum_i \sum_j P_{i,j} \log P_{i,j}, \quad P_{i,j} = \bar{S}_{i,j} / \sum_k \bar{S}_{i,k}, \quad \text{subject to } \bar{S}_{i,j} > T_s, \tag{7}$$

where $\bar{S}_{i,j}$ represents the cardinality of the set $S_{i,j}$, $T_s$ is the threshold value for the minimum number of required points, and $B = \{B_{I,G}, B_{I,L}, B_{O,G}, B_{O,L}\}$ is the set of four adjustable dilation kernels. The optimal dilation kernels, $B^* = \left\{B^*_{I,G}, B^*_{I,L}, B^*_{O,G}, B^*_{O,L}\right\}$, are estimated by solving (7) for each frame. Note that both level-1 and level-2 entropy are considered in (7) to ensure that the task sets are balanced at all hierarchical levels. Moreover, to restrict the solution space of (7), dilation kernels are defined as odd-valued $N$x$N$ all-ones matrices, e.g., $3 \times 3$ or $5 \times 5$ kernels, and the maximum size of the adopted dilation kernel is set to $11 \times 11$ to prevent sampling points that are far from the edge pixels. Since the number of possible dilation kernels is finite, the solution of (7) can be obtained by using exhaustive search. An example that illustrates the four edge regions extracted by using the proposed method is depicted in Fig. 3.

As shown in Fig. 3 (b), most of the boundary regions of the objects on the table are effectively detected. Detailed structures, such as the keyboard and the mouse, can be detected using the Canny edge detector as depicted in Fig. 3 (c). Moreover, the band of occluding-edge regions is wider than that of Canny-edge regions to balance the set size, which is automatically adjusted using (7). The intersection regions, as marked in yellow color in Fig. 3 (d), are discarded in the meta-training phase.

**Algorithm 2** Edge-based meta-ICP algrithm

**Require:** $T_s$, $T_\theta$, $T_p$: threshold values
**Require:** $\alpha$, $\beta$: step size hyper-parameters
**Require:** $N$, $M$: number of iterations
Initialize the transformation vector $\theta$ as zero vector
Get the optimal dilation kernels $B^*$ by (7) and then get the partitioned task sets $\{S_{0,1}, S_{1,1}, S_{1,2}, S_{2,1}, S_{2,2}, S_{2,3}, S_{2,4}\}$ using $B^*$
//meta-training iterations
**for** $t = 1$ to $(N–M)$ **do**
  **for** $i = 1$ to 4 **do**
    Sample $K$ datapoints $D_i = \{p_k, q_k\}$, $k = 1 \sim K$, from $S_{2,i}$
    Compute the adapted parameter vector:
    $$\theta'_i = \theta^t + \Delta\theta_i = \theta^t + f_\theta\left(D_i, \theta^t\right)$$
    Sample another $K$ datapoints $D'_i = \{p_k, q_k\}$, $k = 1 \sim K$, from $S_{2,i}$
  **end for**
  Update $\theta^{t+1} = \theta^t + \alpha \cdot \sum_i f_\theta\left(D'_i, \theta_i''\right)$
  Update $C$ using (11)
  **If** $C == 1$ **then**
    $M = t$
    **break end for**
//use the root-level task set $S_{0,1}$ to perform meta-testing iterations
**for** $t = (N–M+1)$ to $N$ **do**
  **for** $j = 1$ to $T$ **do**
    Sample $K$ datapoints $D^j_t = \{p_k, q_k, \ldots\}$, $k = 1 \sim K$, from $S_{0,1}$
    Evaluate $\Delta\theta^j_t = \left(D^j_t, \theta^t\right)$
  **end for**
  Update $\theta^{t+1} = \theta^t + \beta \cdot \sum_j \Delta\theta^j_t$
**end for**

### B. DYNAMIC MODEL ADAPTATION

At the beginning of a meta-training iteration, the datapoints for each task are sampled from the corresponding edge regions. The adapted parameter vector for task $i$, calculated using the sampled datapoints and the shared parameter vector, is defined as:

$$\theta'_i = \theta^t + \Delta\theta_i = \theta^t + f_\theta\left(D_i, \theta^t\right), \tag{8}$$

where $\theta^t$ is the shared parameter vector for the $t$-th iteration, which is the transformation vector introduced in II-A. $\Delta\theta_i$ is the incremental transformation for task $i$, which is obtained using (3). $D_i = \{p_k, q_k, \ldots\}$ is the sampled datapoints (also known as the support set) for task $i$. $p_k$ and $q_k$ are the $k$-th sampled correspondence pair.

Since the traditional ICP tends to become trapped in local minima, adopting different sampled point sets might result in different locally optimal solutions. By jointly updating the shared parameter vector across all tasks based on the adapted parameters, edge regions that belong to different tasks can be

equally treated. Therefore, the impact of individual spurious estimation, caused by certain noisy edge regions, can then be suppressed. Note that once the adapted parameters become trapped, the task can no longer contribute to the shared parameters until next iteration.

To increase the estimation efficiency, this work investigated the updating strategy of the model parameters in MAML algorithm and introduced a dynamic model adaptation scheme to disturb the trapped tasks. Specifically, the shared parameter vector is updated using the following equation:

$$\theta^{t+1} = \theta^t + \alpha \cdot \sum_i f_\theta \left( D_i', \theta_i'' \right), \tag{9}$$

where $D_i' = \{p_k, q_k, \ldots\}$ is the re-sampled datapoints (also known as the query set) for task $i$, and $\alpha$ is the step size. The symbol $\theta_i''$ represents the adapted parameter vector for task $i$, which is dynamically adjusted as follows:

$$\theta_i'' = \theta^t + \Delta\theta_j, j = \begin{cases} \underset{k}{\text{argmax}} \left\| \Delta\theta_k \right\|_1, & \left\| \Delta\theta_i \right\|_1 < T_\theta \\ i, & others, \end{cases} \tag{10}$$

where $T_\theta$ is the threshold value used for evaluating the parameter trapping condition. The parameter vector with maximal L1 norm is suggested as the alternative adapted parameter vector. Moreover, to save the computational power for performing meta-testing, a convergence checking criterion is adopted to early terminate the meta-training phase. That is, the meta-training process is terminated once the average point-point distances for all tasks are acceptably small. The termination enabling signal $C$ is expressed as:

$$C = \begin{cases} 1, & \max\left(d_1, d_2, d_3, d_4\right) < T_p \\ 0, & others, \end{cases} \tag{11}$$

where $d_i$ and $T_p$ are the average point-point distance of the support set $D_i$ and the distance threshold value, respectively. In the meta-testing phase, the root-level task set $S_{0,1}$, constructed by all task sets used in meta-training, is adopted for pursuing a globally optimized result with respect to the introduced edge features. Moreover, in each meta-testing iteration, this work performed repeated task sampling and then aggregated the estimations to further improve the estimation robustness. The complete edge-based meta-ICP algorithm involving both meta-training and meta-testing is summarized in Algorithm 2.

## IV. EXPERIMENTAL RESULTS

The TUM RGB-D dataset [21] was used for evaluating the proposed edge-based meta-ICP algorithm. The dataset includes image sequences recorded from the Kinect V1 along with the corresponding ground-truth camera poses captured from a motion capture system. Two commonly used error metric were adopted for evaluating the resulting performance. The relative pose error (RPE) was used for measuring the translation drift over a predefined interval which was set as one second (corresponding error unit: cm/s). The absolute trajectory error (ATE) was used for measuring the absolute distances between the estimated and the ground-truth trajectory. The root-mean-square error (RMSE) of the RPEs and the ATEs was adopted for evaluating the resulting accuracy for each sequence. Moreover, to remove the impact of initial prediction and fairly compare different strategies, all experiments were conducted using frame-to-frame estimation without initial pose prediction. The experiments were conducted by using a PC with Intel Core i5-4590 CPU @ 3.3GHz and 32GB memory, and the developed algorithms were coded in un-optimized python code on Ubuntu 18.04 to perform the desired simulation.

### A. EDGE FEATURE COMPARISONS

The experiments were performed by first constructing the complementary task sets obtained by extending the extracted occluding edges and Canny edges, and then partitioning those edges into task sets. After that, the performance of adopting the extended edge features (root-level set $S_{0,1}$) and the original edge features (occluding edges and Canny edges) was compared by using traditional ICP algorithm. Moreover, to relax the influence of Kinect noise for achieving better alignment quality, each correspondence was weighted based on the noise model introduced in [9]. The experimental results of performing 30 iterations with 1000 points for each iteration are summarized in Table 1 where the best value for each test sequence is marked in bold. Because the proposed algorithm adopted random sampling, we reported the average results and the worst results from 30 repeated experiments.

The occurrence and impact of unstable estimations in a test sequence can be evaluated using the ATE metric because the pose estimation errors of the past frames are accumulated. As shown in Table 1, sampling points in the extended edge regions instead of adopting the original edge points can overall improve ATE by 19.7% on the average. As known, camera pose estimation for fast motion sequences are challenging [13]. For the two fast motion sequences fr1/desk and fr1/desk2, adopting the proposed extended edge features can greatly reduce the ATE by 28% on the average. In particular, the worst-case ATE among the 30 repeated experiments is significantly improved by more than 50% for the desk2 sequence.

### B. META-ICP EVALUATION

To evaluate the performance of the proposed meta-ICP algorithm, experiments were done using the same set of extended edge features as those obtained in Table 1. The total number of iterations was set as $N = 30$, including $M = 3$ for meta-testing iterations. Note that the task sets were jointly optimized during the meta-training phase. The size of a support set and a query set was set as $K = 1000$. To guarantee that all task sets have enough points for sampling, the minimum number of points for each task set, $T_s$, was set as 2000. The experimental results of employing the proposed algorithm are listed on the right side of Table 2. For ease of comparison with the results of applying the traditional ICP

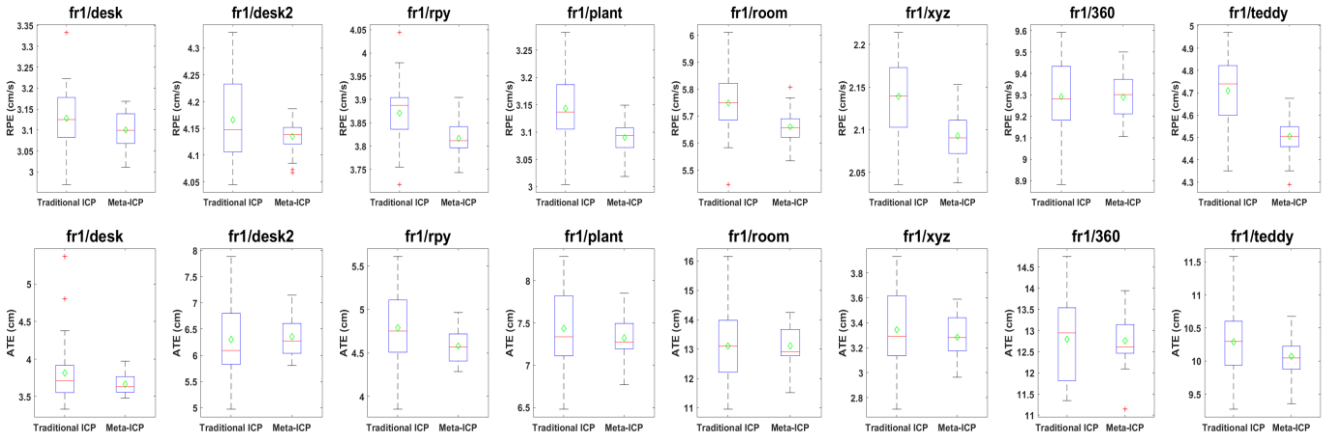


**FIGURE 4.** Boxplots of traditional ICP and meta-ICP algorithms employing extended edge features. Figures in the top row and bottom row are RPE plots and ATE plots, respectively. The average value of each boxplot is indicated by the green diamond.

**TABLE 1.** Performance comparsions when applying two different edge features.

| Average (Worst) | Original edge features | | Extended edge features | | | |
|---|---|---|---|---|---|---|
| | RPE (cm/s) | ATE (cm) | RPE (cm/s) | ATE (cm) | RPE reduction | ATE reduction |
| fr1/ desk | 3.19 (3.33) | 5.29 (6.98) | **3.13** (**3.33**) | **3.81** (**5.37**) | -1.9% (0.0%) | -27.9% (-23.1%) |
| fr1/ desk2 | 4.51 (5.29) | 8.77 (16.86) | **4.17** (**4.33**) | **6.30** (**7.87**) | -7.7% (-18.1%) | -28.2% (-53.2%) |
| fr1/ rpy | 4.07 (4.27) | 6.10 (7.39) | **3.87** (**4.04**) | **4.79** (**5.61**) | -4.9% (-5.2%) | -21.6% (-24.1%) |
| fr1/ plant | 3.30 (3.46) | 8.38 (9.67) | **3.14** (**3.28**) | **7.44** (**8.29**) | -4.9% (-5.1%) | -11.2% (-14.3%) |
| fr1/ room | **5.64** (**5.87**) | 14.63 (17.35) | 5.75 (6.01) | **13.11** (**16.16**) | 1.9% (2.3%) | -10.4% (-6.9%) |
| fr1/ xyz | 2.16 (2.25) | 4.05 (4.64) | **2.14** (**2.21**) | **3.35** (**3.93**) | -1.0% (-1.6%) | -17.4% (-15.2%) |
| fr1/ 360 | 9.68 (10.14) | **12.77** (**14.52**) | **9.29** (**9.59**) | 12.79 (14.75) | -4.0% (-5.4%) | 0.2% (1.6%) |
| fr1/ teddy | 5.75 (6.13) | 17.03 (19.40) | **4.71** (**4.97**) | **10.29** (**11.58**) | -18.0% (-18.9%) | -39.6% (-40.3%) |
| Overall | 4.79 (5.09) | 9.63 (12.10) | **4.52** (**4.72**) | **7.73** (**9.20**) | -5.5% (-7.3%) | -19.7% (-24.0%) |

**TABLE 2.** Performance comparisons between traditional ICP and Meta-ICP algorithms.

| Average (Worst) | Traditional ICP | | Meta-ICP | | | |
|---|---|---|---|---|---|---|
| | RPE (cm/s) | ATE (cm) | RPE (cm/s) | ATE (cm) | RPE reduction | ATE reduction |
| fr1/ desk | 3.13 (3.33) | 3.81 (5.37) | **3.10** (**3.17**) | **3.66** (**3.97**) | -0.9% (-4.9%) | -3.9% (-26.0%) |
| fr1/ desk2 | 4.17 (4.33) | **6.30** (7.87) | **4.14** (**4.19**) | 6.35 (**7.15**) | -0.7% (-3.3%) | 0.8% (-9.4%) |
| fr1/ rpy | 3.87 (4.04) | 4.79 (5.61) | **3.82** (**3.90**) | **4.58** (**4.96**) | -1.4% (-3.5%) | -4.3% (-11.5%) |
| fr1/ plant | 3.14 (3.28) | 7.44 (8.29) | **3.09** (**3.15**) | **7.32** (**7.85**) | -1.7% (-4.1%) | -1.5% (-5.3%) |
| fr1/ room | 5.75 (6.01) | **13.11** (16.16) | **5.66** (**5.81**) | 13.11 (**14.25**) | -1.5% (-3.4%) | 0.0% (-11.8%) |
| fr1/ xyz | 2.14 (2.21) | 3.35 (3.93) | **2.09** (**2.15**) | **3.29** (**3.59**) | -2.2% (-2.8%) | -1.8% (-8.7%) |
| fr1/ 360 | 9.29 (9.59) | 12.79 (14.75) | **9.29** (**9.50**) | **12.76** (13.94) | 0.0% (-1.0%) | -0.3% (-5.5%) |
| fr1/ teddy | 4.71 (4.97) | 10.29 (11.58) | **4.50** (**4.68**) | **10.07** (**10.67**) | -4.4% (-5.9%) | -2.1% (-7.8%) |
| Overall | 4.52 (4.72) | 7.73 (9.20) | **4.46** (**4.57**) | **7.64** (**8.30**) | -1.4% (-3.3%) | -1.2% (-9.8%) |

algorithm, the information in Table 1 is duplicated on the left side of Table 2.

As shown in Table 2, meta-ICP outperforms traditional ICP for almost all the test cases. Traditional ICP only performs slightly better (<1%) than meta-ICP for desk2 and room sequences in terms of the average ATE. Moreover, by adopting the proposed meta-ICP algorithm, the occurrences of unstable estimations can be effectively suppressed, thus improving the worst ATE. The experimental results reveal that the worst ATE can be reduced by more than 25% for desk, and 9% for desk2, rpy, and room using the proposed algorithm. Since the worst-case performance is quite close to the average-case performance when using the proposed algorithm, this indicates a much reliable and robust pose estimation than the traditional one.

For more detailed analysis, the boxplot was adopted to illustrate the error distribution over 30 repeated experiments. The resulting RPE and ATE boxplots of all sequences are depicted in Fig. 4, where the outlier data are denoted by red crosses in the boxplots. As can be observed from the figures, the outlier probability of applying the proposed meta-ICP is much smaller than that of using the traditional ICP for desk and rpy. Moreover, adopting the proposed one can effectively reduce the interquartile range (IQR) of both RPE and ATE for all sequences. The error distributions exhibit that the proposed one possesses good generalization ability for the evaluated sequences.

Table 3 shows the RPE and ATE comparisons with related edge-based visual odometry (VO) algorithms [13], [25] and a typical feature-based approach ORB-SLAM 2 [22]. Specifically, Canny-VO [25] presented two alternatives, namely the approximate nearest neighbor fields (ANNF) and the oriented nearest neighbor fields (ONNF), to replace the distance transform for improving the registration efficiency and accuracy. REVO [13] evaluated two machine learning

**TABLE 3. RPE and ATE comparisons with related edge-based VOs.**

| RPE/ (ATE) | ORB2 VO [22] | Canny VO [25] | | REVO [13] | | | | Proposed algorithm |
|---|---|---|---|---|---|---|---|---|
| | | ANNF | ONNF | Canny [16] | SE [14] | HED RGB [15] | HED RGBD [15] | |
| fr1/desk | 6.18 (9.09) | 7.5 (21.2) | 3.1 (4.4) | 4.80 (10.54) | 7.80 (18.65) | 7.31 (16.72) | 6.79 (12.67) | **3.10** (**3.66**) |
| fr1/desk2 | 6.53 (10.09) | 15.6 (38.1) | 13.1 (18.7) | 7.47 (12.96) | 7.06 (16.87) | 6.15 (11.89) | 7.24 (15.16) | **4.14** (**6.35**) |
| fr1/rpy | **3.22** (8.09) | 6.3 (20.5) | 3.4 (4.7) | 4.05 (12.30) | 3.55 (8.93) | 3.52 (7.87) | 3.67 (7.57) | 3.82 (**4.58**) |
| fr1/plant | 4.22 (7.23) | 5.0 (13.3) | 3.6 (5.9) | 3.58 (5.67) | **3.06** (7.30) | 3.08 (**4.36**) | 3.27 (4.43) | 3.09 (7.32) |
| fr1/room | 7.08 (20.28) | 22.3 (62.1) | **4.2** (24.2) | 5.83 (26.77) | 4.82 (30.59) | 5.78 (34.40) | 6.03 (35.86) | 5.66 (**13.11**) |
| fr1/xyz | **1.47** (**0.88**) | 4.5 (13.7) | 1.9 (4.3) | 4.23 (13.31) | 3.20 (9.01) | 4.81 (14.17) | 4.75 (12.34) | 2.09 (3.29) |
| Overall | 4.78 (9.28) | 10.20 (28.15) | 4.88 (10.37) | 4.99 (13.59) | 4.92 (15.23) | 5.11 (14.90) | 5.29 (14.67) | **3.65** (**6.39**) |

edge detectors [14], [15] and estimated camera poses by minimizing the total edge distances between the source and target frames. Note that for comparison with the proposed algorithm, only the frame-to-frame tracking results in REVO [13] are considered. As shown in Table 3, the proposed algorithm achieves the best RPE for the two fast motion sequences, namely desk and desk2, and the lowest ATE for four out of six sequences. Overall, the experimental results demonstrates that the proposed one can obtain the lowest RPE and ATE on the average as compared to the related works. Leveraging the concept of meta-learning technique [17], the proposed edge-based meta-ICP algorithm can be effectively adopted in different application scenarios according to our experiments. Finally, although the meta-ICP algorithm may take more computational effort than the traditional ICP due to the demand for multiple task estimations, these tasks can be easily dispatched to different processing units to reduce the overall computational time.

## V. CONCLUSION

This paper proposed an edge-based meta-ICP algorithm for reliable camera pose estimation. To construct useful and complementary tasks for performing meta-training, this paper first presented a task set partition algorithm based on the extracted occluding edges and Canny edges. An entropy-based cost function was also introduced to determine the optimal partition of the inner and outer regions for these two edge types. Moreover, a dynamic model adaptation scheme was employed to adaptively adjust the adapted parameters for disturbing the trapped tasks. Experimental results have shown that the worst-case RPE and ATE can be effectively suppressed by adopting the proposed meta-ICP algorithm. Finally, as compared to the traditional ICP algorithm, the proposed one can achieve a smaller IQR of error distributions, indicating much reliable pose estimations. A lower average RPE and ATE can also be achieved using the proposed algorithm in comparison with the related works.

## REFERENCES


[1] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

[2] N. Gelfand, L. Ikemoto, S. Rusinkiewicz, and M. Levoy, "Geometrically stable sampling for the ICP algorithm," in *Proc. Int. Conf. 3-D Digit. Imag. Modeling (3DIM)*, Oct. 2003, pp. 260–267.

[3] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 1991, pp. 2724–2729.

[4] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proc. Int. Conf. 3-D Digit. Imag. Modeling (3DIM)*, May 2001, pp. 145–152.

[5] D. Simon, "Fast and accurate shape-based registration," Ph.D. dissertation, Robot. Inst., Carnegie Mellon Univ., Pittsburgh, PA, USA, 1996.

[6] J. Guehring, "Reliable 3D surface acquisition, registration and validation using statistical error models," in *Proc. Int. Conf. 3-D Digit. Imag. Modeling*, May 2001, pp. 224–231.

[7] C. Choi, A. J. B. Trevor, and H. I. Christensen, "RGB-D edge detection and edge-based registration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2013, pp. 1568–1575.

[8] L. Bose and A. Richards, "Fast depth edge detection and edge based RGB-D SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 1323–1330.

[9] C. V. Nguyen, S. Izadi, and D. Lovell, "Modeling kinect sensor noise for improved 3D reconstruction and tracking," in *Proc. Int. Conf. 3D Imag., Modeling, Process., Vis. Transmiss. (3DIMPVT)*, Oct. 2012.

[10] J. J. Tarrio and S. Pedre, "Realtime edge-based visual odometry for a monocular camera," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 702–710.

[11] P. F. Felzenszwalb and D. P. Huttenlocher, "Distance transforms of sampled functions," *Theory Comput.*, vol. 8, no. 1, pp. 415–428, 2012.

[12] M. Kuse and S. Shen, "Robust camera motion estimation using direct edge alignment and sub-gradient method," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 573–579.

[13] F. Schenk and F. Fraundorfer, "Robust edge-based visual odometry using machine-learned edges," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 1297–1304.

[14] P. Dollár and C. L. Zitnick, "Fast edge detection using structured forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 8, pp. 1558–1570, Aug. 2015.

[15] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1395–1403.

[16] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[17] K. Pulli, "Multiview registration for large data sets," in *Proc. Int. Conf. 3-D Digit. Imag. Modeling (3DIM)*, 1999, pp. 160–168.

[18] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. Int. Conf. Mach. Learn*, vol. 70, Aug. 2017, pp. 1126–1135.

[19] C. Finn, K. Xu, and S. Levine, "Probabilistic model-agnostic meta-learning," in *Proc. NeurIPS*, 2018, pp. 1–14.
[20] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," 2018, *arXiv:1803.02999*. [Online]. Available: http://arxiv.org/abs/1803.02999
[21] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2012, pp. 573–580.
[22] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.
[23] R. Vuorio, S. H. Sun, H. Hu, and J. J. Lim, "Multimodal model-agnostic meta-learning via task-aware modulation," in *Proc. NeurIPS*, 2019, pp. 1–22.
[24] K. Xu, S. Levine, and C. Finn, "Unsupervised learning via meta-learning," in *Proc. ICLR*, 2019, pp. 1–24.
[25] Y. Zhou, H. Li, and L. Kneip, "Canny-VO: Visual odometry with RGB-D cameras based on geometric 3-D–2-D edge alignment," *IEEE Trans. Robot.*, vol. 35, no. 1, pp. 184–199, Feb. 2019.
[26] A. Raghu, M. Raghu, S. Bengio, and O. Vinyals, "Rapid learning or feature reuse? Towards understanding the effectiveness of MAML," in *Proc. ICLR*, 2020, pp. 1–21.
[27] R. L. Gutierrez and M. Leonetti, "Information-theoretic task selection for meta-reinforcement learning," in *Proc. NeurIPS*, 2020, pp. 1–11.
[28] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of kinect depth data for indoor mapping applications," *Sensors*, vol. 12, no. 2, pp. 1437–1454, Feb. 2012.

**CHUN-WEI CHEN** received the B.S. degree in electrical engineering from National Sun Yat-sen University, Kaohsiung, Taiwan, in 2010, and the M.S. degree in electrical engineering from National Cheng Kung University, Tainan, Taiwan, in 2012, where he is currently pursuing the Ph.D. degree.

His research interests include machine learning, computer vision, point cloud processing, and visual SLAM.

**JONAS WANG** received the M.S. degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA.

He was a Manager of notebook graphics engineering with Trident Micro Systems, Arden, NC, USA. He has been a Deputy Director of video IP development with Himax Technologies, Inc., Tainan, Taiwan, since 2007. He has over 20 years of experience in the semiconductor industry with extensive experience in system-on-a-chip design for video processing and notebook graphics.

**MING-DER SHIEH** (Member, IEEE) received the B.S. degree in electrical engineering from National Cheng Kung University, Tainan, Taiwan, in 1984, the M.S. degree in electronic engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1986, and the Ph.D. degree in electrical engineering from Michigan State University, East Lansing, MI, USA, in 1993. From 1993 to 2002, he was a Faculty Member with the Department of Electronic Engineering, National Yunlin University of Science and Technology (NYUST), Douliu, Taiwan. From 1999 to 2002, he was the Department Chairman with NYUST. Since 2002, he has been with the Department of Electrical Engineering, National Cheng Kung University. From 2010 to 2014, he was the Deputy General Director of Information and Communications Research Laboratories, Industrial Technology Research Institute (ITRI), Taiwan. From 2014 to 2017, he was the Department Chairman with National Cheng Kung University, where he is currently a Full Professor. His current research interests include very large scale integration (VLSI) design and testing, VLSI for signal processing, and digital communication. He was a technical committee member in several international conferences. He received the Teaching Award from NYUST, in 1998. He was the Program Co-Chair and General Co-Chair of the Asian Test Symposium, in 2004 and 2009, respectively, and the Chairman of the Tainan Chapter of the IEEE Circuits and Systems Society, from 2009 to 2010. From 2010 to 2012, he served as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS and the Lead Guest Editor of a special issue of *Computers and Electrical Engineering* journal, in 2012.