# Contrastive Learning-based Identification and Multi-view-based 3D Positioning

Khoa Anh Ngo, Jihoon Moon, and Byonghyo Shim
INMC, Department of Electrical and Computer Engineering, Seoul National University, Korea
Email: {ngoak, jhmoon}@islab.snu.ac.kr, bshim@snu.ac.kr

***Abstract*—Recently, computer vision-aided positioning techniques have emerged as a promising approach to achieve precise positioning in wireless communications. However, maintaining a constant view of the user equipment (UE) is a challenging task due to occlusions. In this paper, we propose a multi-view positioning technique, referred to as *contrastive learning-based identification and multi-view-based positioning (CLI-MVP)*, for multi-user communication systems. To be specific, we detect and identify UEs in multi-view images and estimate their positions using the triangulation technique.**

## I. Introduction

In recent years, the emergence of 6G communication systems has spurred the development of various position-based services (PBSs) such as wireless communications, autonomous driving, and tactile internet that require accurate positioning [1]–[3]. However, since positioning techniques in 5G NR primarily rely on radio frequency, their positioning performance depends heavily on the communication environment and does not meet the stringent requirements of PBSs.

In this paper, we propose a positioning technique using multi-view images for multi-user communication systems. The proposed technique, referred to as *contrastive learning-based identification and multi-view-based positioning* (CLI-MVP), consists of three steps: 1) UE detection, 2) UE identification, and 3) UE positioning. That is, we first detect UEs using a deep learning (DL) based object detector. Then, we identify the same UE across multi-view images based on the similarity of visual features, which are learned using *contrastive learning*. Finally, the positions of the identified UEs are estimated using triangulation.

## II. Multi-view Positioning via Contrastive Learning

In this section, we describe in detail the process of estimating the positions of UEs using multi-view images.

### A. Object Detection

We consider a multi-view vision-aided system consisting of $N$ cameras, resulting in a set of images $\mathcal{I} = \{I_f \in \mathbb{R}^{h \times w \times 3}\}_{f=1}^{N}$, where $I_f$ is an image from the $f$-th camera, $w$ and $h$ are the width and height of an image, respectively. The position of a UE $\hat{\mathbf{p}}_\mathbf{k}$ can be estimated by analyzing its projections in the multi-view images. To obtain the projections of UEs in the $f$-th image, we employ a DL-based object detector that finds out $K$ bounding boxes, where $b_{k,f}$ is a bounding box of $k$-th object. The bounding box $b_{k,f} = (x^I_{k,f}, y^I_{k,f}, w^I_{k,f}, h^I_{k,f})$ contains the center cartesian coordinate $(x^I_{k,f}, y^I_{k,f})$ and the width and height $(w^I_{k,f}, h^I_{k,f})$ of the selected rectangular area in the image plane.

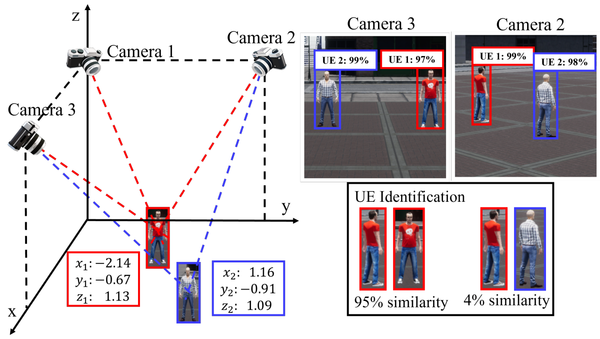

Fig. 1: Illustration of the multi-view positioning system. The UEs are detected in each image and then paired based on the visual similarity, and the position of each UE is estimated as the intersection of light rays.

### B. UE Identification

To identify UE across multi-view images, we pair a bounding box from the $f$-th camera with a bounding box from the $g$-th camera based on their visual features, such as clothes and body shapes. Specifically, a DL-based network, namely ResNet, receives an input image $I_{k,f} \in \mathbb{R}^{w^I_{k,f} \times h^I_{k,f} \times 3}$, cropped from the image $I_f$ according to the bounding box $b_{k,f}$, and extracts visual features $\mathbf{a}_{k,f} \in \mathbb{D}$, where $D$ is the feature dimension.

To guide the DL-based network to learn the visual features of the same UE across multi-view images, we implement contrastive learning that identifies the features that are available in all cameras. Formally, given visual features from the $f$-th camera and the $g$-th camera, we maximize the cosine similarity if the bounding boxes are from the same UE and minimize otherwise,

$$\mathcal{L}(b_{k,f}, b_{j,g}) = (1 - 2 * \mathbb{1}(k,j))\, s_{(k,f),(j,g)}, \tag{1}$$

where $\mathbb{1}(k,j)$ is an indicator function that returns 1 if $k$ and $j$ are the same UE, and 0 otherwise, and $s_{(k,f)(j,g)}$ is the similarity measurement of visual features $\mathbf{a}_{k,f}$ and $\mathbf{a}_{j,g}$ of $k$-th UE in $f$-th view and $j$-th UE in $g$-th view, respectively, using cosine similarity,

$$s_{(k,f)(j,g)} = \frac{\mathbf{a}^{\mathrm{T}}_{k,f}\mathbf{a}_{j,g}}{\|\mathbf{a}_{k,f}\|\|\mathbf{a}_{j,g}\|}. \tag{2}$$

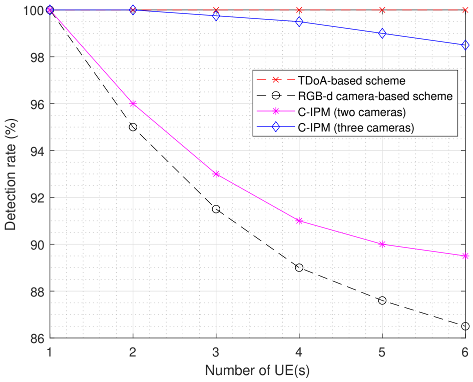

Fig. 2: Detection rate vs. the number of UEs.

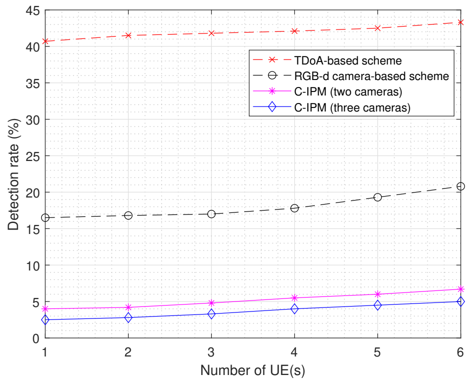

Fig. 3: Distance error vs. the number of UEs.

To pair bounding boxes from the $f$-th camera $\mathcal{B}_f = \{b_{k,f}\}$ with bounding boxes from the $g$-th camera $\mathcal{B}_g = \{b_{j,g}\}$, we find the bipartite matching between bounding boxes with the lowest matching cost,

$$\hat{\sigma} = \underset{\sigma \in \Delta_K}{\arg\max} \sum_{k}^{K} \left(1 - s_{(k,f)(\sigma(k),g)}\right), \tag{3}$$

where $\alpha(k)$ is a mapping function that maps bounding box indices from the $f$-th image $f$ to the $g$-th image. The optimal assignment can be found using the Hungarian algorithm.

### C. UE Positioning

Using the paired bounding boxes from all cameras, we triangulate the 3D position that results in the centroid pixels in each image. We assume the camera position $\mathbf{o}_f = [o_{f,x}\ o_{f,y}\ o_{f,z}]$, the camera rotation $\mathbf{R}_f \in \mathbb{R}^{3\times3}$, the camera intrinsic parameter matrix $\mathbf{K}_f \in \mathbb{R}^{3\times3}$ and are known. The light ray that projects the $k$-th UE onto the $f$-th camera is represented as a linear line $\mathbf{l}_{k,f} = \mathbf{o}_f + t\mathbf{v}_{k,f}$ in 3D space where $\mathbf{v}_{k,f} = [x_{k,f}\ y_{k,f}\ 1]\mathbf{K}_f^{-1}\mathbf{R}_f^{-1}$ is the direction vector.

Using the reconstructed light rays, we estimate the position of UEs by finding their midpoints, as the rays might not intersect. We can express the light ray as

$$(\mathbf{I} - \mathbf{V}_{k,f})(\hat{\mathbf{p}}_k - \mathbf{o}_f) = \mathbf{0}, \tag{4}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{V}_{k,f} = \mathbf{v}_{k,f}(\mathbf{v}_{k,f}^{\mathrm{H}}\mathbf{v}_{k,f})^{-1}\mathbf{v}_{k,f}^{\mathrm{H}}$ is the projection matrix onto the directional vector $\mathbf{v}_{k,f}$. Collecting (4) from cameras $f = 1, \cdots, N$, we can construct an overdetermined linear system with $N$ equations. The system can be expressed simply as $\mathbf{G}\hat{\mathbf{p}}_k = \mathbf{b}$, where

$$\mathbf{G} = \begin{bmatrix} \mathbf{I} - \mathbf{V}_{k,1} \\ \vdots \\ \mathbf{I} - \mathbf{V}_{k,N} \end{bmatrix}, \mathbf{b} = \begin{bmatrix} (\mathbf{I} - \mathbf{V}_{k,1})\mathbf{o}_1 \\ \vdots \\ (\mathbf{I} - \mathbf{V}_{k,N})\mathbf{o}_N \end{bmatrix}. \tag{5}$$

Then, the LS solution of the linear system can be easily computed as $\hat{\mathbf{p}}_k = (\mathbf{G}^{\mathrm{T}}\mathbf{G})^{-1}\mathbf{G}^{\mathrm{T}}\mathbf{b}$.

## III. Simulation Results

In this section, we examine the performance of the proposed multi-view positioning technique in terms of detection rate and positioning error. To train and evaluate the positioning performance of CLI-MVP, we consider people as UEs and generate a dataset where UEs are randomly placed in the $7 \times 7$m area using Unity Game Engine. For comparison, we consider a single-view positioning technique and a TDoA-based positioning technique.

In Fig. 2, we illustrate the detection rate of various positioning methods as a function of the number of UEs. Firstly, we observe that CLI-MVP achieves 99.3% whereas the detection rate of the single-view positioning scheme is 88%. The detection rate of the single-view positioning scheme degrades drastically when the number of UEs increases as one UE occludes another. Secondly, the detection rate of CLI-MVP increases with the number of cameras, 99%, and 90% using three and two cameras, respectively. The TDoA-based method trivially achieves a 100% detection rate using the transmitted signals from all UEs.

In Fig. 3, we plot the positioning error of various positioning schemes as a function of a number of UEs. The TDoA-based method achieves 40cm with high variance due to the variation of environment. In contrast, the single-view method utilizing an RGB-D camera can achieve positioning error at 15 cm but with a low detection rate. CLI-MVP, however, achieves a near-perfect detection rate and precise positioning at a 6cm positioning error. That is, multi-view positioning methods utilizing two and three cameras achieve positioning errors of 5.5 and 5.4 cm, respectively. The more precise estimated positions result in a triple achievable data rate compared to the TDoA-based scheme.

## References

[1] Z. Chen, C. Han, Y. Wu, L. Li, C. Huang, Z. Zhang, G. Wang, and W. Tong, "Terahertz wireless communications for 2030 and beyond: A cutting-edge frontier," *IEEE Commun. Mag.*, vol. 59, no. 11, pp. 66–72, 2021.

[2] Z. Hou, C. She, Y. Li, D. Niyato, M. Dohler, and B. Vucetic, "Intelligent communications for tactile internet in 6G: requirements, technologies, and challenges," *IEEE Commun. Mag.*, vol. 59, no. 12, pp. 82–88, 2021.

[3] H. Chen, H. Sarieddeen, T. Ballal, H. Wymeersch, M.-S. Alouini, and T. Y. Al-Naffouri, "A tutorial on terahertz-band localization for 6G communication systems," *IEEE Commun. Surveys Tuts.*, 2022.