ViFiT: Reconstructing Vision Trajectories from IMU and Wi-Fi Fine Time Measurements

Bryan Bo Cao[*†], Abrar Alali[‡§], Hansi Liu[¶], Nicholas Meegan[¶], Marco Gruteser[¶], Kristin Dana[¶], Ashwin Ashok[♯], Shubham Jain[†]

[†]Stony Brook University, [‡]Old Dominion University, [§]Saudi Electronic University,
[¶]Rutgers University, [♯]Georgia State University

[†]{boccao,jain}@cs.stonybrook.edu, [‡]{aalal003}@odu.edu, [¶]{hansiiii,gruteser}@winlab.rutgers.edu,
[¶]{njm146}@scarletmail.rutgers.edu, [¶]{kristin.dana}@rutgers.edu, [♯]{aashok}@gsu.edu

ABSTRACT

Tracking subjects in videos is one of the most widely used functions in camera-based IoT applications such as security surveillance, smart city traffic safety enhancement, vehicle to pedestrian communication and so on. In computer vision domain, tracking is usually achieved by first detecting subjects, then associating detected bounding boxes across video frames. Typically, frames are transmitted to a remote site for processing, incurring high latency and network costs. To address this, we propose *ViFiT*, a transformer-based model that reconstructs vision bounding box trajectories from phone data (IMU and Fine Time Measurements). It leverages a transformer's ability of better modeling long-term time series data. *ViFiT* is evaluated on Vi-Fi Dataset, a large-scale multimodal dataset in 5 diverse real world scenes, including indoor and outdoor environments. Results demonstrate that *ViFiT* outperforms the state-of-the-art approach for cross-modal reconstruction in LSTM Encoder-Decoder architecture *X-Translator* and achieves a high frame reduction rate as 97.76% with IMU and Wi-Fi data.

CCS CONCEPTS

• **Computer systems organization** → *Sensor networks*; • **Computing methodologies** → **Tracking**; **Object detection**; **Reconstruction**.

KEYWORDS

Transformer, Multimodal Learning, Multimodal Reconstruction, IMU, Object Detection, Tracking, Efficient Video System

ACM Reference Format:
Bryan Bo Cao[*†], Abrar Alali[‡§], Hansi Liu[¶], Nicholas Meegan[¶], Marco Gruteser[¶], Kristin Dana[¶], Ashwin Ashok[♯], Shubham Jain[†]. 2023. *ViFiT*: Reconstructing Vision Trajectories from IMU and Wi-Fi Fine Time Measurements. In *3rd ACM MobiCom Workshop on Integrated Sensing and Communication Systems for IoT (ISACom) (ISACom '23), October 6, 2023, Madrid, Spain.* ACM, Madrid, Spain, 6 pages. https://doi.org/10.1145/3615984.3616503

1 INTRODUCTION

Tracking of human subjects in camera videos plays a key role in many real world applications, such as security surveillance, accident prevention, and traffic safety. State-of-the-art visual trackers rely on visual information from cameras but fail in scenarios with limited visibility, caused by poor light conditions, occlusion of the objects being tracked, or in out-of-view regions. Moreover, cameras installed for surveillance or other applications typically send all the frames from the footage to a remote location for processing - imposing constraints on network bandwidth. While it is possible to downsample the number of frames sent over the network, it is not desirable for applications such as tracking, that may require fine-grained information and can be negatively impacted by missing frames. This creates an inherent tradeoff between the requirement to preserve network bandwidth and the fidelity of the camera video. Limiting the number of frames can preserve network bandwidth, however, it will lead to missed image frames and thus tracking errors. To address this issue and the limitations of tracking based on vision only, prior works have leveraged complementary modalities, such as raw sensory or meta data from phone and wireless signals, primarily through *multimodal association* [1, 12, 13]. However, these approaches fail in tracking in the absence of image frames.

To address this gap, we propose *ViFiT*, a system that can reconstruct a human subject's motion trajectory in camera video footage by leveraging motion sensor data from the subject's phone. Specifically, we capture inertial measurement unit (IMU) and Wi-Fi Fine Time Measurements (FTM) readings from the subjects' phones to reconstruct their vision trajectories. By leveraging lightweight modalities from the phone, we can ensure continuity in tracking information even when the camera frames are missing or the subjects are occluded or out of the camera view. Our proposed approach also identifies the minimum number of camera frames required for vision trajectory reconstruction. In the future, *ViFiT* can also serve as an adaptive downsampling technique to reduce the amount of camera data transmitted over the network.

Contributions. In summary, our contributions are:

- We design and develop *ViFiT*, a novel multimodal transformer-based model to reconstruct vision tracklets from phone domain data, including IMU and FTM data.
- We develop a novel Intersection over Union (IoU) based metric called *Minimum Required Frames* (MRF) that jointly captures both the quality of reconstructed bounding boxes and the lower bound of frames required for reconstruction.
- *ViFiT* achieves an MRF of 37.75 in all 4 outdoor scenes on average, outperforming the best baseline method with an MRF of 53 (Δ =

ISACom '23, October 6, 2023, Madrid, Spain

ACM ISBN 979-8-4007-0364-5/23/10…$15.00
https://doi.org/10.1145/3615984.3616503

15.25). *ViFiT* uses only 2.24% of frames from the videos, demonstrating its effectiveness to reconstruct accurate bounding boxes (IoU>0.5) without requiring all the frames in a video stream.

We conducted a study on the amount of data in network transmission in different modalities. Our empirical results show that by using *ViFiT* and multiple modalities, we can significantly reduce the amount of information (more than 99%) in network transmission.

Challenges and Approach. As illustrated in Fig. 1, *ViFiT* proposes an approach to reconstructing the tracklets (series of tracking coordinates) for each subject from their phone's IMU and Wi-Fi data. We encounter several challenges: (1) ***Coordinate frame transformation***: the phone and camera have different reference coordinate frames and therefore translating from one to the other requires a coordinate frame alignment. A naive extension of IONet [3] that converts trajectories in a map representation to image coordinates will fail due to new challenges in generalizing to various cameras especially when their camera parameters are unknown; (2) ***Multimodal fusion***: the phone data includes raw IMU sensor readings and WiFi FTM values (distance from the access point), which have to be associated with camera image frames through a unified deep learning tracking model; (3) ***IMU cumulative drift***: trajectories computed using IMU data are known to drift over time, potentially increasing errors in the reconstructed vision trajectories.

Keeping in line with the state-of-the-art trackers using transformer-based models, in this paper we investigate, experiment, and evaluate a transformer model with careful designs for vision trajectories reconstruction tasks in the images by using minimal image frames and multiple modalities of phone IMU and FTM data.

To speed up computation, researchers have tried image compression, resolution reduction, or dropping frames. These approaches, however, reduce image quality or remove necessary information completely. As a result, tracking performance is degraded. Our approach differs by reducing the video sampling rate while keeping the integrity of an entire frame. However, as the video stream is downsampled, we lose the subject's fine-grained movements across consecutive frames. Thus, the detected trajectories from a downsampled stream will likely be error-prone.

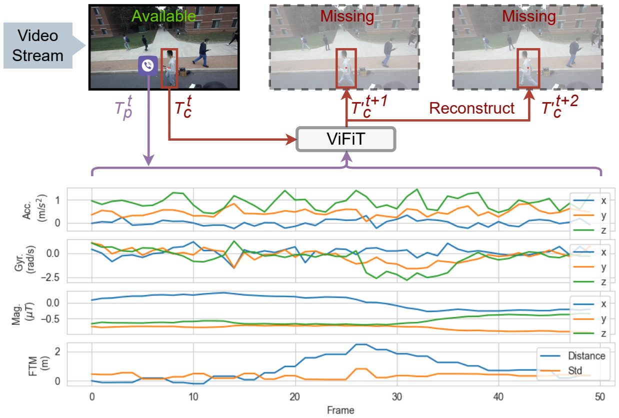


Figure 1: Task formulation: *ViFiT* reconstructs bounding boxes in missing frames by using complementary phone data – accelerations, gyroscope, magnetometer readings and wireless FTM data.

2 RELATED WORK

Vision-based Detection & Tracking Recently transformer-based models have been deployed for a wide variety of visual tasks, such as image classification [5], object detection [2] and tracking [4]. Common vision benchmarks include COCO [10] for object detection, MOT [14] for tracking. However, these are for vision-only evaluation when all the frames in a complete video are available. Visual trajectory reconstruction datasets include BIWI [15] and [8] but lack phone modality. We hereby use the Vi-Fi dataset [11] [12] that includes vision, IMU and FTM data.

Multimodal Learning and Fusion. The closest works to our research from multimodal learning aspect include Vi-Fi [12], ViTag [1] and ViFiCon [13], which focus on multimodal association. Our task differs by reconstructing vision trajectories using phone sensor data. In addition, compared to these recurrent or convolutional models, we take a step forward to explore transformer's capacity to learn long-term time series data from IMU and multimodal fusion.

3 SYSTEM OVERVIEW

3.1 Data Preprocessing

We utilize the large-scale multimodal dataset from Vi-Fi [12]. Vi-Fi dataset consists of RGB-D (depth) visual data captured by a ZED-2 stereo camera, wireless data by communication between Google Pixel 3a phones and a Google Nest Wi-Fi access point next to the camera, as well as IMU accelerometer, gyroscope, magnetometer sensor readings from the phone. It covers a wide range of scenes both indoors and outdoors, totaling 142K frames in 89 sequences, each of which lasts around 3 minutes. At most 5 subjects are holding phones communicating with the access point and 11 detections in one scene. All participants walk in an unconstrained fashion.

Camera Data. We follow the procedure in ViTag [1] to generate trajectories (referred to as *tracklets* in the rest of the paper) using the StereoLabs ZED tracker on the RGB-Depth camera data. Tracklets are typically short because subjects move out of the field of view of the camera frequently. Tracklets from camera data (T_c) are represented as a time series sequence of bounding boxes (BBX). Each bounding box is represented as:

$$BBX = [x, y, d, w, h], \quad T_c \in \mathbb{R}^{WL \times 5} \tag{1}$$

where x and y are the coordinates of the centroid of the bounding box, d is the centroid's depth measurement, and w and h are the bounding box width and height, respectively. WL is the window's length which is also the number of frames in a window.

Phone Data. To preprocess the smartphone data, we use 3 types of measurements from the time series IMU data. We extract the 3-axis accelerometer data acc, 3-axis gyroscope $gyro$, and magnetometer data mag. These measurements from the time series IMU data are concatenated as a vector:

$$T_i^t = [acc; gyro; mag], \quad T_i \in \mathbb{R}^{WL \times 9} \tag{2}$$

Additionally, we use phones' FTM measurements at time t which is defined as:

$$T_f^t = [r, std], \quad T_f \in \mathbb{R}^{WL \times 2} \tag{3}$$

where r represents the estimated range, or distance from phone to WiFi access point, while std represents the standard deviation calculated in a single RTT burst.

In the context of our work, we use *modality* to refer to one type of data such as bounding boxes, IMU readings, or FTM data, while we use *domain* to refer to the source, such as camera or smartphone. Thus, vision tracklets (T_c) belong to the camera domain, and IMU and FTM data belong to the phone domain (T_p):

$$T_p = [T_i; T_f]. \tag{4}$$

Multimodal Synchronization. All the modalities are synchronized before feeding them to the model. We use Network Time Protocol (NTP) on the devices to synchronize the camera and phone data. The sampling rate for camera frames is 30 FPS, for IMU readings is 100 Hz, and 3-5 Hz for FTM. Each phone datapoint finds the closest timestamp of a camera frame.

Normalization. By default, we apply normalization on IMU data following the same way as LIMU-BERT [17].

3.2 ViFiT Design

In this section, we describe the details of *ViFiT*. The workflow is illustrated in Fig. 2 (b). We employ the main transformer backbone inspired by LIMU-BERT [17] with a few key modifications: (1) we implement three separate independent encoders to learn modality-specific features; (2) intermediate representations are concatenated to fuse multiple modality information; and (3) residual connections between encoder and decoder are removed such that decoder purely depends on the fused multimodal representations.

ViFiT [1] consists of three encoders for multiple modalities and one vision decoder. The objective of the encoders is to learn multimodal representations from vision, IMU and wireless data FTM, e.g. IMU Encoder is responsible for capturing motion information from accelerations, rotations and orientations while FTM Encoder focuses on wireless data. We keep encoders identical for all modalities for the simplicity to extend new modalities such as RF in the future work. In the next step, representations are fused by concatenation, based on which the Vision Decoder reconstructs the bounding boxes.

Encoder. Each encoder comprises $B = 4$ stacks of Multi-head Self-attention (MSA), Projection and Feed Forward layers, with residual connections and Layer Normalization in between. It takes in a tracklet T_m as input and projects it to a higher dimensional space:

$$X = Proj(T_m) = A \times T_m \tag{5}$$

where A is a matrix with dimension $H_{dim} \times D_m$ and H_{dim} is the hidden space dimension larger than modality m's feature dimension D_m (e.g. $D_c = 5, D_i = 9$ and $D_f = 2$). We set $H_{dim} = 72$ by default as it yields the best overall performance shown in later experimental sections. We implement the Projection layer *Proj(·)* by a linear layer. The objective is to expand the low dimensional input feature space D_m to a larger one to learn richer implicit features X_m for modality m. Since data in different scenes have different distributions, it can lead to unstable training. Therefore, Layer Normalization is applied to stabilize features of instance i from previous Projection Layer:

$$\hat{X}_m^i = LayerNorm(X_m^i) = \frac{X_{m,j}^i - \mu_j}{\sqrt{\sigma_j^2 + \epsilon}}\gamma + \beta \tag{6}$$

where γ and β are the learnable hyperparameters, and ϵ is a small number to avoid numerical instability. The mean and standard deviation across modality m's feature j (jth column of A) are denoted by μ_j and σ_j, respectively. After that, different from recurrent layers of LSTM in the Vi-Fi [12] or ViTag [1] models, we employ positional encoding to learn the order information of modality m and add it into $\hat{X}_m$, followed by a second Layer Normalization.

In the next step, $\hat{X}_m$ enters B core transformer blocks with Multi-head Self-attention (MSA) layers. Scaled Dot-product Attention is used with d_k dimensional queries and keys while values are of dimension d_v, implemented by:

$$MSA(\hat{X}_m) = MSA(\hat{Q}_m, \hat{K}_m, \hat{V}_m) = Concat(head_1, ..., head_h)A_m^O \tag{7}$$

where $A^O \in \mathbb{R}^{hd_v \times d_{model}}$ and a *head* is an attention layer:

$$Attention(\hat{X}_m) = Attention(\hat{Q}_m, \hat{K}_m, \hat{V}_m) = softmax(\frac{\hat{Q}_m \hat{K}_m^T}{\sqrt{d_k}})\hat{V}_m \tag{8}$$

where $\hat{Q}_m = \hat{K}_m = \hat{V}_m = \hat{X}_m$ for a modality's self-attention. More numbers of heads (h) allow for learning different representations. In this implementation we set $h = 4$, $d_k = d_v = H_{dim} = 72$ and $d_{model} = H_{dim} \times h = 72 \times 4 = 288$.

The Position-wise Feed Forward layer (denoted as *FFN(·)* and Feed Forward in Fig. 2) is implemented by two linear transformations of dimension $H_{dim} = 72$ and $F_{dim} = 144$, respectively. Following [17], Gaussian Error Linear Unit (GELU) [7] is utilized as the activation function between two layers.

The following functions constitute a transformer block:

$$M^b = LayerNorm(MSA(\hat{X}_m^{b-1}) + \hat{X}_m^{b-1}) \tag{9}$$

$$P^b = LayerNorm(Proj(M^b) + M^b) \tag{10}$$

$$\hat{X}_m^b = LayerNorm(FFN(P^b) + P^b) \tag{11}$$

where $b - 1$ denotes a previous block. The final representation of modality m from encoder is denoted as X_m' shown in Fig. 2.

Decoder. The decoder is implemented for the vision modality only. Concatenated multimodal representations are fed into the decoder, which is comprised of a projection *Proj(·)*, GELU activation *GELU(·)* and Layer Normalization *LayerNorm(·)*, followed by a linear prediction head *Pred(·)* of dimension H_{dim}:

$$X'_{fused} = Concat(X'_c, X'_i, X'_f) \tag{12}$$

$$\hat{X}_{fused} = Proj(X'_{fused}) \tag{13}$$

$$T'_c = Pred(LayerNorm(GELU(\hat{X}_{fused}))) \tag{14}$$

3.3 Training

Loss Functions. The task is formulated as a regression problem, which maps camera domain information T_c^0 with first frame only and phone domain data T_i and T_f to a continuous space in T'_c. Therefore, we employ Mean Squared Error (MSE) as the default loss function by:

$$L_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(T_c^i, {T'_c}^i)^2 \tag{15}$$

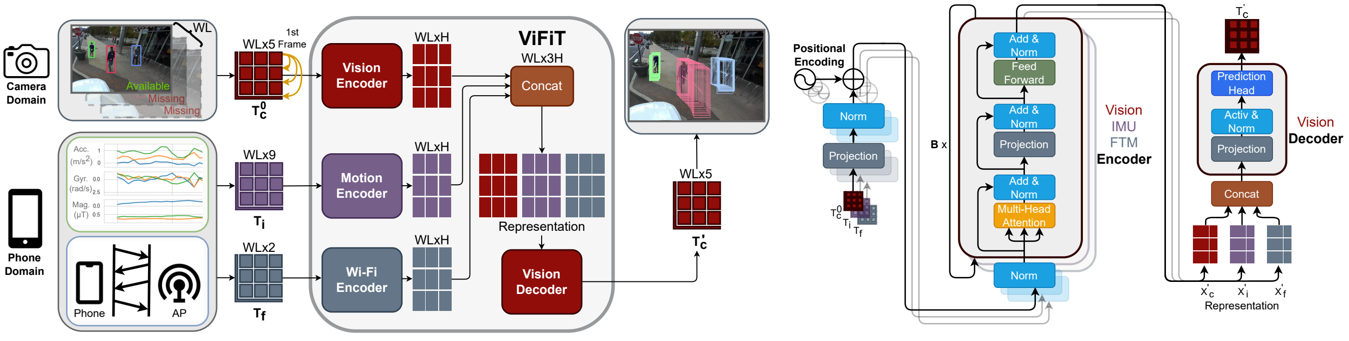
where T_c is ground truth (GT) and T'_c is the reconstructed tracklet in a window, i is BBX index in a tracklet and N is the number of training samples.

To train the model for better bounding box estimation, we further leverage Distance-IoU (DIoU) loss inspired by Zheng et al. [18]:

$$L_{DIoU} = \frac{1}{N}\sum_{i=1}^{N}\left(1 - IoU(T_c^i, {T'_c}^i) + \frac{\rho^2(T_c^i, {T'_c}^i)}{(s^i)^2}\right) \tag{16}$$

where $\rho(\cdot)$ is the Euclidean Distance between the centroids of ith BBX in T_c^i and ${T'_c}^i$ and s^i denotes the diagonal length of the smallest enclosing box that covers those two bounding boxes.

[1] Code is available at https://github.com/bryanbocao/vifit. Dataset can be downloaded at https://sites.google.com/winlab.rutgers.edu/vi-fidataset/home.



(a) ViFit System Overview

(b) ViFit Architecture

Figure 2: (a) *ViFiT* System Overview. *ViFiT* consists of multimodal Encoders for (T_c^0, T_i and T_f) to extract features and a Vision Decoder to reconstruct the complete visual trajectory of T_c' for the missing frames in a window with length WL. Note T_c^0 denotes a vision tracklet with first frame only and H denotes representation dimension. (b) Vi-Fi Transformer (*ViFiT*) Architecture. *ViFiT* is comprised of multimodal Vision, IMU and FTM Encoders depicted on the left side in parallel displayed with various opacity, as well as a Vision Decoder on the right. Information flow starts from the bottom left corner, where each tracklet for one modality (T_c^0, T_i or T_f) is fed into its own Encoder independently, including B blocks of transformer modules with Multi-head Self-attention (MSA). In the next step, Encoders generate multimodal representations, fused by concatenation (X_c', X_i', X_f') and are fed into the Vision Decoder to output bounding boxes (T_c') in missing frames.

4 EVALUATION

4.1 Baseline Methods

We evaluate our approach by comparison against alternative methods. Baseline methods are categorized into two categories: (1) **traditional handcrafted methods**, including *Broadcasting* (BC), *Pedestrian Dead Reckoning* (PDR) [16], and *Kalman Filter* [9], as well as (2) **deep learning methods** that includes *X-Translator* [1]. The details of the baselines are described as follows:

***Broadcasting* (BC).** The first frame detections are broadcasted through the rest of missing frames.

***Pedestrian Dead Reckoning* (PDR) [16].** PDR is the process of estimating the current position using previous estimates by IMU sensors in the North-East-Down (NED) coordinates. For fair benchmarking, we follow the procedure in Vi-Fi [12] and ViTag [1] to construct a trajectory with 2D points of $(\hat{x}^t, \hat{y}^t)$ at time t in a 2D map from IMU readings, and convert them into feet center points in an image. To generate other bounding box parameters (w, h and d) in the image coordinate system, we learn two linear functions f_1, f_2 to regress bounding box widths and heights, and a quadratic function for depths from the camera horizontal position y.

***Kalman Filter* (KF) [9].** Kalman filter is widely used in localization and state estimation. Accurate tracking can be attributed to the weight adjustment between measurements and state prediction errors by the importance variable Kalman Gain. We adopt a kalman filter to estimate a person's bounding box position in an image.

***X-Translator* (X-T) [1].** A multimodal LSTM network in encoder-decoder architecture from ViTag. We follow the training procedure from the publicly released code. Different from the reconstruction path from vision to phone tracklets ($T_c \rightarrow T_p'$) in ViTag, we utilize the other reconstruction path from phone data to vision tracklets ($T_p \rightarrow T_c'$) to perform the same task in this paper. We also feed T_c^0 with only the first frame into the model.

4.2 Evaluation Metrics

Evaluation protocol is consistent across different methods. Specifically, a reconstruct method (RM) is a function that takes in (1) $T_c^0 \in \mathbb{R}^{1\times5}$ with bounding box detections in the first frame only, (2) $T_i \in \mathbb{R}^{WL\times9}$ and (3) $T_f \in \mathbb{R}^{WL\times2}$ across all frames in a window as input and outputs the reconstructed bounding boxes $T_c' \in \mathbb{R}^{WL\times5}$:

$$T_c' = RM(T_c^0, T_i, T_f) \tag{17}$$

For each metrics described in later subsections, a reconstruct method (RM)'s output T_c' is compared to the reference of ground truth (GT) $T_{c(GT)}$ in a window.

Intersection Over Union (IoU). IoU is referred to as the Jaccard similarity coefficient or the Jaccard index, computed by the area of overlap between the reconstructed bounding box and the GT at *ith* frame divided by their union.

Average Precision (AP). AP is commonly used in object detection calculated by the ratio of True Positives (TP) over all the positives, where a threshold τ of IoU is given to determine a TP. IoU greater than 0.5 is generally considered good.

Unless otherwise specified, each of the aforementioned scores is computed per subject per frame.

Minimum Required Frames (MRF). We introduce MRF as the main metric to measure the smallest number of frames needed for an RM to reconstruct good bounding boxes in a video stream system. Existing common metrics of IoU and AP only focus on the quality of reconstructed bounding boxes in each frame independently, but fails to capture frame-related characteristics. To measure the upper bound of the number of frames a video can be dropped without compromising the reconstruction performance below a certain threshold, we propose a novel IoU-based metric coined *Minimum Required Frames* (MRF). The main intuition of MRF is to capture the minimum number of frames required in order for a method to continuously reconstruct decent bounding boxes

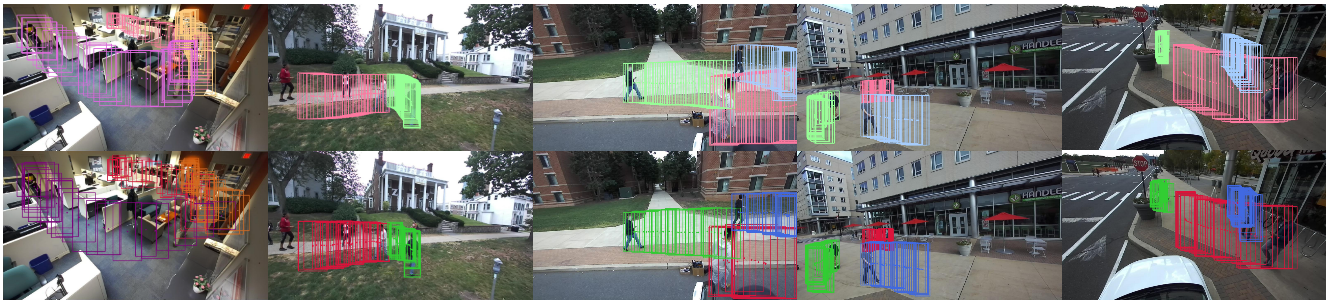


Figure 3: Samples of reconstructed vision tracklets T_c' and ground truths GT decorated in lighter (1st row) and darker colors (2nd row). Indoor scene is shown in the 1st column while outdoor scenes are displayed from the 2nd to the 5th columns. By visual comparison between T_c' and GT, *ViFiT* is capable of generating decent bounding boxes for missing frames.

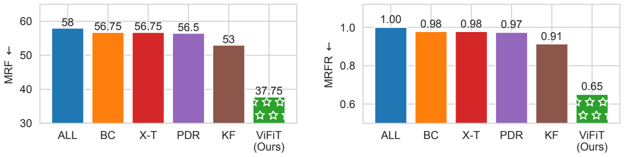
in missing frames. MRF is computed in sliding window way while only the first frame in a window is available. Smaller MRF indicates better reconstructions. Given a video stream V consisting of F frames, window length WL that satisfies $W > 2$ and IoU threshold τ, the algorithm computes MRF for the reconstruct method RM.

Since the total number of frames in a video stream F varies, MRF will also be changed even if the distribution preserves for the same RM. We hereby introduce *Minimum Required Frame Ratio* (MRFR) which is defined as MRF divided by the total number of windows:

$$MRFR = \frac{MRF}{W} \tag{18}$$

4.3 Overall Performance

Our main result is presented in Fig. 4. In summary, *ViFiT-30F* (*ViFiT* trained on 30-frame windows) trained by DIoU loss yields the lowest *Minimum Required Frames* (MRF) of 37.75 across all 4 outdoor scenes, exceeding the second best method *KF* with 53 by 15.25, demonstrating the effectiveness of our approach in reconstructing bounding boxes for missing frames by fusing phone motion and wireless data. Evaluation is done using window length $WL = 30$ and stride $WS = 29$. Longer window lengths cover more complicated trajectories such as making turns. Therefore, we evaluate the model *ViFiT-30F* with a longer window length than *ViFiT-10F* in the following study.



(a) Minimum Required Frames (MRF)

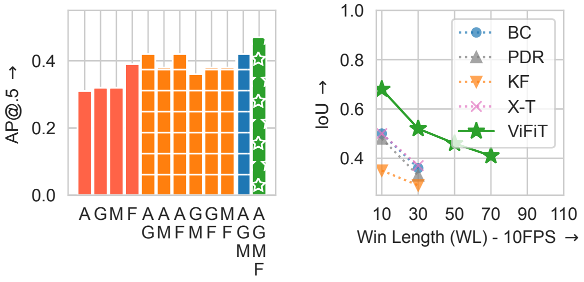
(b) Minimum Required Frames Ratio (MRFR)

Figure 4: Main results: comparison of our system *ViFiT-30F* (DIoU) against baselines evaluated by *Minimum Required Frames* (MRF) in (a) and MRFR (Ratio) in (b) for all 4 outdoor scenes with window length $WL = 30$, stride $WS = 29$. ALL: average of total number of processed windows in one sequence, BC: *broadcasting*, X-T: *X-Translator-30F*, PDR: *Pedestrian Dead Reckoning*, KF: *Kalman Filter*. Results demonstrate the effectiveness of *ViFiT-30F* to reduce frames with only $\frac{37.75}{1683} = 2.24\%$ of video frames needed to reconstruct good bounding boxes (IoU = $0.55 > 0.5$).

To interpret the result of *ViFiT-30F* in Fig. 4, *RM* processes 58 windows ($1 + 58 \times 29 = 1683$ frames) in one scene shown by "ALL", out of which in each of the 37.75 windows *ViFiT-30F* queries bounding boxes of the first frame from video on average, resulting in 2.24% frames used. In other words, *ViFiT-30F* has generated good bounding boxes for the rest of missing frames, accounting for $1 - 2.24\% = 97.76\%$ of a video. Note the average IoU and AP@.5 of these predictions are 0.55 and 0.56 while IoU > 0.5 is generally considered good. This result has demonstrated the effectiveness of our system by exploiting other modalities from phone domain, including IMU and wireless data to save video data. Compared to IoU or AP, we highlight the practical benefit of MRF that it tells a practitioner a quantitative number to determine the minimum frames (e.g. for saving network transmission) required for reconstruction.
Continuous Reconstructed Trajectory Length Interval. On average, *ViFiT-30F* is able to continuously reconstruct decent bounding boxes in 1.55 windows, resulting in 45.01 frames, which correspond to 4.5 seconds with frame rate of 30 FPS. *ViFiT-30F* is capable of reconstructing a continuous trajectory with a maximum length of 87 frames. Samples are visualized in Fig. 3. For each window, reconstructed bounding boxes are shown in Row 1. Compared to the GT in Rows 2, we can see that *ViFiT* can generate decent bounding boxes in a window using only a single frame.

4.4 System Analysis

In this section, we analyze the system from various perspectives.
Phone Feature Ablation Study. We conduct an ablation study on accelerometer (A), gyroscope (G), magnetometer (M) and FTM (F) data shown in Fig. 5 (a). Overall, combining all features yields the best performances with IoU, AP@.5 and AP@.1 of 0.5, 0.47 and 0.82, respectively. A single feature results in lower scores (light red), while any combination generally assists in better reconstruction. F is more useful when combining all A, G and M, demonstrating the design choice of phone features in our system.
Benchmark on Window Length (WL). We conduct benchmarking on various methods by varying window lengths (WL) in outdoor scenes in Fig. 5 (b). Observe that most existing methods fail to produce desirable bounding boxes (IoU > 0.5) when only the first frame is available. Overall, *ViFiT* outperforms all baselines.
Analysis on Vision-only Object Detector. We evaluate the state-of-the-art vision model's detection when all frames are available. By the time we conduct the experiment, YOLOv5 [6] is the most



(a) Phone Ablation Study (b) Methods for Various WL

Figure 5: (a): Phone Feature Ablation Study on *Vi-Fi-Former-30F* in all 5 scenes. A: Acceleration, G: Gyroscope Angular Velocity, M: Magnetometer Reading, F: FTM. With all phone features (stars in green), *Vi-Fi-Former-30F* achieves the best performance. (b): Top right is better. *ViFiT* achieves higher AP with longer window lengths compared to other baselines.

Model	AP@.5 ↑	FR ↓	Modality	#Params (M)
YOLOv5m	0.91	100 %	Vision	21.2
YOLOv5n	0.90	100 %	Vision	1.9
ViFiT-10F	0.82	**2.25 %**	Vision, **IMU, FTM**	**0.15**
ViFiT-30F	0.56	**2.24 %**	Vision, **IMU, FTM**	**0.15**

Table 1: Analysis with Vision-only model YOLOv5 in outdoor scenes. RF: Frame Ratio, n: nano, m: medium. *ViFiT-10F* and *ViFiT-30F* are able to preserve decent bounding boxes with IoU 0.71 and 0.55 (>0.5 good), resulting in AP@.5 of 0.82 and 0.56, respectively.

recent model. Results are shown in Table 1. Overall, *ViFiT-10F* and *ViFiT-30F* preserve decent bounding boxes, achieving an AP@.5 as 0.82 and 0.56 in the outdoor scenes, compared to 0.91 and 0.90 by YOLOv5m (medium) and YOLOv5n (nano), respectively. *ViFiT-10F* and *ViFiT-30F* achieve an IoU of 0.71 and 0.55 while IoU>0.5 is generally considered as good. Although our model requires additional IMU and FTM data, the large visual data reduction with only an extremely small portion of 2.24% frames needed in a video demonstrates it as a decent trade-off. Last but not least, our model is lightweight (0.15M #Params) that does not bring much overhead to existing systems, compared to 1.9M and 21.2M for YOLOv5 nano and medium.

5 CONCLUSION

In this paper we designed *ViFiT*, a system that hosts a transformer-based deep learning model to generate tracking information for missing frames in a video. Our work showed that by using strategically selected small number of video frames along with phone’s IMU data as well as Wi-Fi FTM can achieve high tracking accuracy. In particular, *ViFiT* uniquely is able to *reconstruct* the motion trajectories of human subjects in videos even if the corresponding video frames were not made available or considered lost for processing. To properly evaluate the video characteristics of the system, we propose novel IoU-based metrics *Minimum Required Frames* (MRF) and *Minimum Required Frames Ratio* (MRFR) for a Camera-GPU system. *ViFiT* achieves 0.65 MRFR, significantly lower than the second best method *Kalman Filter* of 0.91 and the state-of-the-art LSTM-based model *X-Translator* of 0.98, resulting in an extremely large amount of frame reduction of 97.76%. Through extensive experiments we demonstrated that *ViFiT* is capable of tracking the target with high accuracy, making it applicable in real world scenarios.

6 ACKNOWLEDGEMENT

This research has been supported by the National Science Foundation (NSF) under Grant Nos. CNS-2055520, CNS-1901355, CNS-1901133.

REFERENCES

[1] Bryan Bo Cao, Abrar Alali, Hansi Liu, Nicholas Meegan, Marco Gruteser, Kristin Dana, Ashwin Ashok, and Shubham Jain. 2022. ViTag: Online WiFi Fine Time Measurements Aided Vision-Motion Identity Association in Multi-person Environments. In *2022 19th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 19–27.

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer, 213–229.

[3] Changhao Chen, Xiaoxuan Lu, Andrew Markham, and Niki Trigoni. 2018. Ionet: Learning to cure the curse of drift in inertial odometry. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[4] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. 2021. Transformer tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 8126–8135.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[6] Glenn Jocher et. al. 2021. *ultralytics/yolov5: v6.0 - YOLOv5n 'Nano' models, Roboflow integration, TensorFlow export, OpenCV DNN support.* https://doi.org/10.5281/zenodo.5563715

[7] Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415* (2016).

[8] Ronny Hug, Stefan Becker, Wolfgang Hübner, and Michael Arens. 2021. Quantifying the complexity of standard benchmarking datasets for long-term human trajectory prediction. *IEEE Access* 9 (2021), 77693–77704.

[9] Qiang Li, Ranyang Li, Kaifan Ji, and Wei Dai. 2015. Kalman filter and its application. In *2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS)*. IEEE, 74–77.

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 740–755.

[11] Hansi Liu. 2022 [Online]. Vi-Fi Dataset. https://sites.google.com/winlab.rutgers.edu/vi-fidataset/home

[12] Hansi Liu, Abrar Alali, Mohamed Ibrahim, Bryan Bo Cao, Nicholas Meegan, Hongyu Li, Marco Gruteser, Shubham Jain, Kristin Dana, Ashwin Ashok, et al. 2022. Vi-Fi: Associating Moving Subjects across Vision and Wireless Sensors. In *2022 21st ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 208–219.

[13] Nicholas Meegan, Hansi Liu, Bryan Cao, Abrar Alali, Kristin Dana, Marco Gruteser, Shubham Jain, and Ashwin Ashok. 2022. ViFiCon: Vision and Wireless Association Via Self-Supervised Contrastive Learning. *arXiv preprint arXiv:2210.05513* (2022).

[14] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. 2016. MOT16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831* (2016).

[15] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. 2009. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th international conference on computer vision*. IEEE, 261–268.

[16] Boyuan Wang, Xuelin Liu, Baoguo Yu, Ruicai Jia, and Xingli Gan. 2018. Pedestrian dead reckoning based on motion mode recognition using a smartphone. *Sensors* 18, 6 (2018), 1811.

[17] Huatao Xu, Pengfei Zhou, Rui Tan, Mo Li, and Guobin Shen. 2021. LIMU-BERT: Unleashing the Potential of Unlabeled Data for IMU Sensing Applications. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*. 220–233.

[18] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. 2020. Distance-IoU loss: Faster and better learning for bounding box regression. In