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, ¶ Georgia State University

{boccao,jain}@cs.stonybrook.edu, †{aalal003}@odu.edu, ‡{hansiii.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

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM Reference Format:

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

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 =
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.

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.

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.

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 ViFi [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:

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

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:

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

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

$$T_F = [r, std], \quad T_F \in \mathbb{R}^{WL \times 2}$$

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_v) belong to the camera domain, and IMU and FTM data belong to the phone domain (T_p):

\[ T_p = [T_i; T_f]. \]  

(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 detail the design 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 were removed such that decoder purely depends on the fused multimodal representations.

ViFiT comprises 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, for 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 following reasons:

- The objective is to expand the low dimensional input feature space and add it into \( X_m \), followed by a second Layer Normalization.
- In the next step, \( X_m \) 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:

\[
\text{MSA}(X_m) = \text{MSA}(\hat{Q}_m, \hat{K}_m, \hat{V}_m) = \text{Concat}(\text{head}_1, ..., \text{head}_h)A^Q_m
\]

where \( A^Q \in \mathbb{R}^{h d_k \times d_{model}} \) and a head is an attention layer:

\[
\text{Attention}(\hat{X}_m) = \text{Attention}(\hat{Q}_m, \hat{K}_m, \hat{V}_m) = \text{softmax}(\frac{\hat{Q}_m^\top \hat{K}_m}{\sqrt{d_k}})\hat{V}_m
\]

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 \( h = 4, d_k = d_v = D_{dim} = 72 \) 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 \( D_{dim} = 72 \) 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:

\[
M^b = \text{LayerNorm}(\text{MSA}(X_m^{b-1}) + X_m^{b-1})
\]

(9)

\[
F^b = \text{LayerNorm}(\text{Proj}(M^b) + F^b)
\]

(10)

\[
X_m^b = \text{LayerNorm}(\text{FFN}(F^b) + F^b)
\]

(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 \( \text{Proj}(·) \); GELU activation \( \text{GELU}(·) \) and Layer Normalization \( \text{LayerNorm}(·) \), followed by a linear prediction head \( \text{Pred}(·) \) of dimension \( H_{dim} \):

\[
X_{fused}' = \text{Concat}(X_v', X_i', X_f')
\]

(12)

\[
\hat{X}_{fused} = \text{Proj}(X_{fused}')
\]

(13)

\[
T_c' = \text{Pred}(\text{LayerNorm}(\text{GELU}(\hat{X}_{fused})))
\]

(14)

### 3.3 Training

**Loss Functions.** The task is formulated as a regression problem, which maps camera domain information \( T_p' \) 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:

\[
L_{MSE} = \frac{1}{N} \sum_{i=1}^{N} (T_c', T_i')^2
\]

(15)

where \( T_c \) is ground truth (GT) and \( T_i' \) 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-LOU (DLOU) loss inspired by Zheng et al. [18]:

\[
L_{DLOU} = \frac{1}{N} \sum_{i=1}^{N} \left[ 1 - \text{IOU}(T_c', T_c^i) + \rho^2(T_c', T_i^i) \right]
\]

(16)

where \( \rho(·) \) is the Euclidean Distance between the centroids of \( ith \) BBX in \( T_c' \) and \( T_i' \) and \( s' \) denotes the diagonal length of the smallest enclosing box that covers those two bounding boxes.
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 \((x', y')\) 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 \((T_c \rightarrow T'_c)\) 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 \(T'_c\) 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 \(T^0_c, T_i\) with bounding box detections in the first frame only, \(T_i \in \mathbb{R}^{WL \times 9}\) and \(T_f \in \mathbb{R}^{WL \times 2}\) across all frames in a window as input and outputs the reconstructed bounding boxes \(T'_c \in \mathbb{R}^{WL \times 5}\):

\[
T'_c = RM(T^0_c, T_i, T_f)
\]  

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 \(\tau\) 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 fail 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.
Minimum Required Frames Ratio (MRFR) is defined as \( \text{MRF} \) divided by the total number of windows: \( \text{MRFR} = \frac{\text{MRF}}{W} \) (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 \( W_L = 30 \) and stride \( W_S = 29 \). 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.

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
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: FT M. 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.

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.

<table>
<thead>
<tr>
<th>Model</th>
<th>AP@.5</th>
<th>FR</th>
<th>Modality</th>
<th>#Params (M)</th>
</tr>
</thead>
<tbody>
<tr>
<td>YOLOv5n</td>
<td>0.91</td>
<td>100</td>
<td>Vision</td>
<td></td>
</tr>
<tr>
<td>YOLOv5n</td>
<td>0.90</td>
<td>100</td>
<td>Vision</td>
<td>1.9</td>
</tr>
<tr>
<td>ViFiT-10F</td>
<td>0.82</td>
<td>100</td>
<td>Vision, IMU, FT M</td>
<td>0.15</td>
</tr>
<tr>
<td>ViFiT-30F</td>
<td>0.56</td>
<td>100</td>
<td>Vision, IMU, FT M</td>
<td>0.15</td>
</tr>
</tbody>
</table>

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 it 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 MRF, 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


