Efficient Video Redaction at the Edge: Human Motion Tracking for Privacy Protection

HAOTIAN QIAO, University of Michigan-Ann Arbor, Ann Arbor, United States
VIDYA SRINIVAS, University of Washington, Seattle, United States
PETER DINDA, Northwestern University, Evanston, United States
ROBERT DICK, University of Michigan-Ann Arbor, Ann Arbor, United States

Computationally efficient, camera-based, real-time human position tracking on low-end, edge devices would enable numerous applications, including privacy-preserving video redaction and analysis. Unfortunately, running most deep neural network based models in real time requires expensive hardware, making widespread deployment difficult, particularly on edge devices. Shifting inference to the cloud increases the attack surface, generally requiring that users trust cloud servers, and increases demands on wireless networks in deployment venues. Our goal is to determine the extreme to which edge video redaction efficiency can be taken, with a particular interest in enabling, for the first time, low-cost, real-time deployments with inexpensive commodity hardware. We present an efficient solution to the human detection (and redaction) problem based on singular value decomposition (SVD) background removal and describe a novel time-efficient and energy-efficient sensor-fusion algorithm that leverages human position information in real-world coordinates to enable real-time visual human detection and tracking at the edge. These ideas are evaluated using a prototype built from (resource-constrained) commodity hardware representative of commonly used low-cost IoT edge devices. The speed and accuracy of the system are evaluated via a deployment study, and it is compared with the most advanced relevant alternatives. The multi-modal system operates at a frame rate ranging from 20 FPS to 60 FPS, achieves a $wIoU_{0.3}$ score (see Section 5.4) ranging from 0.71 to 0.79, and successfully performs complete redaction of privacy-sensitive pixels with a success rate of 91%–99% in human head regions and 77%–91% in upper body regions, depending on the number of individuals present in the field of view. These results demonstrate that it is possible to achieve adequate efficiency to enable real-time redaction on inexpensive, commodity edge hardware.

CCS Concepts: • **Security and privacy → Privacy protections**; • **Computer systems organization → Embedded software;**

Additional Key Words and Phrases: ERASE (efficient redaction automation system at the edge), human detection and tracking, SVD background subtraction, UWB localization, sensor fusion, real time edge computing, privacy preserving

This work was supported, in part, by NSF under awards CNS-2211509 and CNS-2211508, and in part by the University of Michigan.

Authors' Contact Information: Haotian Qiao, University of Michigan-Ann Arbor, Ann Arbor, Michigan, United States; e-mail: qhaotian@umich.edu; Vidya Srinivas, University of Washington, Seattle, Washington, United States; e-mail: vysri@cs.washington.edu; Peter Dinda, Northwestern University, Evanston, Illinois, United States; e-mail: pdinda@northwestern.edu; Robert Dick, University of Michigan-Ann Arbor, Ann Arbor, Michigan, United States; e-mail: dickrp@umich.edu.

ACM 1539-9087/2025/09-ART120
https://doi.org/10.1145/3762994

ACM Reference Format:
Haotian Qiao, Vidya Srinivas, Peter Dinda, and Robert Dick. 2025. Efficient Video Redaction at the Edge: Human Motion Tracking for Privacy Protection. *ACM Trans. Embedd. Comput. Syst.* 24, 5s, Article 120 (September 2025), 22 pages. https://doi.org/10.1145/3762994

1 Introduction

Privacy is at risk of vanishing as **Internet-of-Things** (**IoT**) sensors proliferate. For example, shopping malls are peppered with cameras, which bring security and market research benefits, but undermine privacy. Malicious actors may use captured video to detect what someone is buying or who they are with and use this information to prioritize targets for blackmail or theft. In theory, one could use remote (cloud) servers for video redaction. However, this requires third-party involvement, and generally involves streaming video data through potentially insecure subsystems, increasing the attack surface. On-device redaction mitigates this problem, decreasing the propagation of sensitive information and reducing the attack surface.

Existing state-of-the-art redaction techniques use computationally intensive, data-driven, deep-learning-based approaches that do not support real-time execution on resource-constrained, inexpensive commodity IoT edge devices. Simply put, using existing techniques would require venues to purchase numerous expensive, high-performance devices, increasing the economic barrier to using privacy-preserving approaches to video analysis. For edge redaction to be scalable, it should be efficient enough to run on the low-cost edge devices that numerically dominate the IoT marketplace.

Our goal is to determine the extreme to which efficiency can be taken for privacy preserving video redaction on inexpensive, commodity edge devices, while preserving high accuracy. Motivated by this problem, we design an approach for efficient redaction on inexpensive, commodity hardware and test it in multiple indoor scenarios. The resulting algorithms dramatically increase the frame rates practical on inexpensive, commodity edge hardware relative to prior approaches, making real-time execution practical, i.e., on-line processing is fast enough to match the data capture rate. Figure 1 illustrates the capabilities of the proposed system, called **Efficient Redaction Automation System at the Edge** (**ERASE**). ERASE takes the video stream and human location data as inputs, estimates and redacts privacy-related regions, and outputs the redacted video. It enables real-time, e.g., at least 10 FPS throughput, privacy-aware redaction on inexpensive ($\leq$ 50 USD) commodity hardware.

Contributions

During the design process, we were motivated by a key constraint: existing neural network models capable of solving the redaction problem do not support real-time execution on inexpensive, commodity edge devices. To address this problem, we developed an efficient redaction algorithm based on the privacy protection problem formulation that fuses video and location data from inexpensive sensors already present in many smartphones and expected to become standard in the future, enabling efficient human detection and redaction. This article makes the following contributions.

(1) A novel problem formulation and constrained privacy-aware optimization objective. This formulation enables the use of video and location data for efficient, iterative redaction region optimization.
(2) An efficient gradient-based privacy-aware redaction region optimization algorithm informed by location data and **singular value decomposition** (**SVD**) of video data. This algorithm is dramatically more efficient than existing deep neural network based redaction approaches,

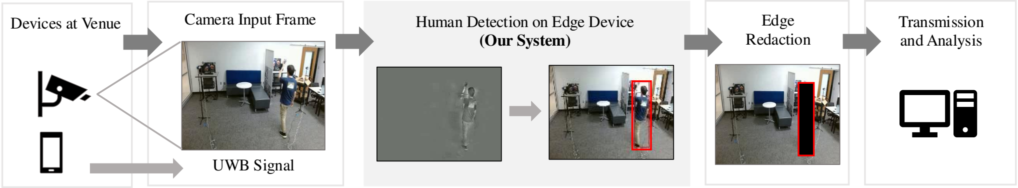


Fig. 1. The proposed system, ERASE, enables real-time visual tracking of people using only inexpensive, commodity IoT devices, and enables the redaction of privacy-relevant data before it leaves the capture device.

enabling, for the first time, real-time operation on inexpensive, commodity edge devices without accelerators or GPUs.

(3) The design and implementation of the above algorithm on a hardware-software system that fuses video data and human position traces obtained from **ultra-wideband** (**UWB**) localization hardware using an ARM Cortex A72 processor.

(4) An efficient approach to mitigate fading, a persistent problem in prior SVD-based background subtraction methods, where stationary target objects are misclassified as background and, consequently, vanish from the extracted foreground (see Section 3.5). This approach integrates UWB-based location information with video information, enabling SVD-based video analysis to be used on resource-constrained edge devices when redacted people (or objects) are sometimes stationary.

(5) A new dataset that provides, for the first time, time-synchronized video and UWB data with various numbers of people in the field of view, under a range of conditions and scenarios.

We evaluated the system embodying these contributions, which is able to perform localization and redaction on (inexpensive) ARM Cortex A72 hardware at over 60 FPS for a single individual, scaling linearly to 20 FPS for five individuals. Previously published redaction approaches measured on the same hardware had a maximum frame rate of 6.26 FPS. Depending on the number of individuals present in the field of view, ERASE successfully performs complete redaction of privacy-sensitive pixels with a success rate of 91%–99% in human head regions and 77%–91% in upper body regions and achieves accuracy ranging from 0.71 to 0.79 in **weighted intersection over union** (**wIoU**), a measure particularly appropriate for redaction system accuracy, which is defined in Section 5.4. These results demonstrate that it is possible to reach adequate efficiency to enable real-time redaction on inexpensive, commodity edge hardware.

2 Related Work

This section describes related work on on-device human detection and tracking methods for privacy protection. To the best of our knowledge, our work is the first to enable efficient vision-based real-time human tracking enabling low-latency, privacy-preserving video redaction on low-cost, commodity edge devices.

Many existing works [13, 15, 20] focus on improving accuracy of UWB indoor localization systems by fusing traces estimated by a vision-based human detection algorithm and UWB system. They use deep neural networks for human detection/pose estimation to track human positions in the 3-D real-world coordinates by projecting human foot locations from the pixel coordinates (the location of pixels in an image) to the real-world coordinates (the location in the predefined 3-D physical coordinates). Finally, they fuse the video traces and UWB traces to improve localization accuracy in 3-D space. In contrast to such systems, our ERASE determines a region (bounding box) in the 2-D camera visual field associated with each person. Most of the algorithmic complexity and contribution of our work stems from determining the best boundaries between privacy-relevant pixels and privacy-neutral pixels in this visual field. Details of our fusion algorithm are described in Sections 4.3 and 4.4.

There are existing works exploring “close-to-data-source” privacy-preserving camera systems, where privacy-relevant pixels are removed before data leave the devices.

PrivacyLens [6] fuses thermal sensing data with RGB camera data using a Jetson Nano embedded GPU or a Titan RTX desktop GPU with the goal of improving redaction accuracy, thereby reducing leakage of privacy-relevant information relative to classical RGB-based approaches. Their approach also improves robustness to changes in lighting conditions compared to conventional RGB approaches. Their focus on thermal camera data and computationally intensive models has several implications. The resulting redacted images can still capture body shape and pose information quite accurately, enabling high inference accuracy on redacted video in several applications, e.g., in exercise repetition counting and fall detection. However, the thermal camera increases the camera node cost by 164 USD and their approach to redaction is relatively computationally intensive, requiring a 100 USD embedded GPU to achieve a frame rate of 8 FPS. The frame rate is also limited by the 8 FPS legal restriction on thermal cameras. In contrast, our focus is on achieving high enough efficiency to enable high frame rates (over 20 FPS) on commodity edge hardware and requires only a commodity RGB camera and inexpensive (less than 60 USD) UWB localization system. Only 4 UWB anchors are needed to form an indoor localization system regardless of the number of cameras deployed in the venue, with the system capable of supporting all camera nodes. In summary, the PrivacyLens paper shows the implications of using thermal sensing data for video redaction and our paper shows the implications of pushing redaction efficiency to the extreme to enable high frame rate implementation using inexpensive commodity hardware. As a result, our paper focuses on detailed algorithm design. Both systems are evaluated on measured data in several real-world use cases.

Opt-In Camera describes a system in which UWB and RGB camera data are fused to enable some individuals to express consent to video capture by carrying tags. This work is currently under review, but a six-page pre-print is accessible [7]. It uses the YOLOv9-Wholebody-with-Wheelchair object tracking package and does not focus on efficiency. A 1,499 USD MacBook Pro with M3 Max is used for evaluation, with which video can be processed at 10 FPS. In contrast, our work focuses on efficiency to enable high frame rates on inexpensive commodity edge hardware. This requires much more attention to low-level algorithm design details.

Our system, ERASE, fuses UWB-based location and video data to enable efficient, accurate, and robust human tracking. This supports (economic) scaling of video systems capable of redaction at the edge. Figure 2 contains the flow chart for ERASE and illustrates the results it produces during each stage of processing. ERASE supports various redaction modes. Our experiments show that the multi-modal system operates at a frame rate between 20 FPS and 60 FPS, achieves a $wIoU_{0.3}$ score (see Section 5.4) ranging from 0.71 to 0.79, and successfully performs complete redaction of privacy-sensitive pixels with a success rate of 91%–99% in human head regions and 77%–91% in upper body regions, depending on the number of individuals present in the field of view. It also achieves over 0.92 recall: the proportion of the privacy-relevant pixels redacted.

3 Problem Definition and Formulation

Problem: Given a series of 2-D arrays containing pixels representing frames captured by stationary cameras as an input, identify and localize human-related pixels in the visual data, and generate the smallest rectangular bounding box, represented as the coordinates of its upper left and lower right corners, i.e., $[(x_{\min}, y_{\min}), (x_{\max}, y_{\max})]$, for each person in the scene that covers all pixels related to the person, constrained to run in real time on inexpensive commodity IoT devices (without GPUs or accelerators). The constraint that the number of boxes must equal the number of people in the

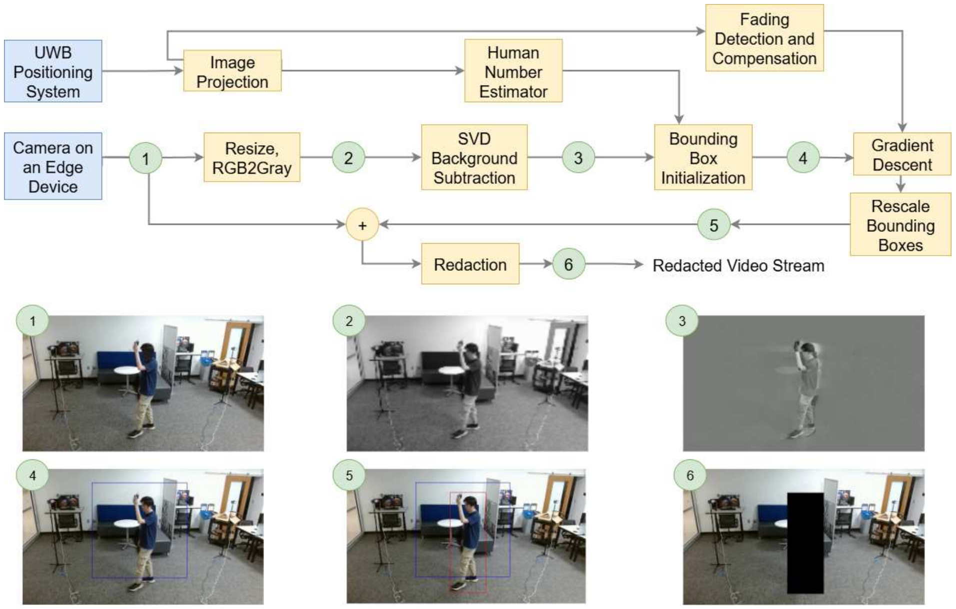


Fig. 2. ERASE system flow chart. Sub-figures 1–6 show the intermediate outputs at each stage of ERASE, illustrating the redaction process.

scene is important. Without it, the problem becomes a segmentation task for which generating a box for every privacy-sensitive pixel would be an optimal solution.

This section gives our problem definition and formulation for vision-based human detection. Our solution builds upon SVD-based background removal [11], a historical term we follow with some reluctance, as it would more accurately be called "SVD-based stationary object removal" in stationary-camera applications. In the classical problem, pixels associated with stationary background and objects are removed. The remaining pixels, which are associated with moving objects, have historically been called "foreground pixels". Our approach uses a quasi-continuous representation of whether each pixel is associated with moving or stationary objects. See Section 3.2 for details about the formulation.

We make the following assumptions:

(1) cameras are stationary, which is common in venues such as supermarkets, shopping malls, and restaurants and
(2) all moving objects are people or extensions of them, e.g., objects held by people. Some violation of this assumption is acceptable, as it results in unnecessary redaction of non-private pixels instead of privacy violation.

Violations of each assumption are accounted for in experimentation, and their influence is reflected in the reported accuracy values.

Note that the computational time of ERASE scales approximately linearly with the number of people in the scene. Detailed analysis will be presented in Section 5.5. One might also imagine a scenario in which human-related pixels dominate the scene, which would cause SVD to misidentify person pixels as background pixels because it breaks the low-rank assumption of SVD background removal: we expect this case to be rare in the applications we are considering, and did not encounter it in our captured video. This situation can be avoided by appropriate selection of camera point of view and focal length.

The following subsections describe SVD background removal, give the mathematical formulation of our privacy-preserving redaction, bounding box optimization, and fading detection algorithms.

3.1 (Static) Background Removal

Our method requires background removal for further processing. The background removal method needs to be **computationally efficient**, **lighting-agnostic**, and **layout-agnostic**. Existing deep learning based background removal techniques are not time efficient on inexpensive edge devices without GPUs or accelerators. Subtracting pre-recorded background is not robust to variable lighting conditions and changes to background layouts. Gaussian mixture model methods like **Mixture of Gaussians** (**MOG**) [22], and SVD background removal are able to address these challenges. MOG can operate in real-time by estimating and updating the background frame by frame, whereas SVD requires processing batches of frames to update the background model. This batching has implications for the worst-case latency of SVD-based approaches: although average throughput may be high, the latency for the first frame in a batch may be relatively poor. MOG is capable of handling dynamic backgrounds, such as gently waving trees, with appropriate parameter tuning. In contrast, SVD provides more accurate background estimates when the camera is stationary, such as in security camera setups, as in our intended applications. Additionally, SVD has fewer hyperparameters to configure, primarily the batch size and the number of singular values used for background estimation. This article focuses on SVD-based background removal. Figure 3 shows example outputs of SVD-based background removal under different lighting conditions, demonstrating its robustness to such variations.

SVD can be used for background removal [16]. During SVD-based background removal, we consider a temporal sequence of grayscale frames of length ζ. See Section 5.3 for a detailed description of the selection of algorithmic parameters. The height and width of each frame are defined as H_s and W_s. Within each window of ζ frames, we flatten each frame into a column vector with $H_s \cdot W_s$ elements. These frames form a batch for SVD removal. In the rest of the paper, a *batch* refers to ζ consecutive frames for background removal. Columns are horizontally concatenated to form a matrix $\mathbf{A}$ with $H_s \cdot W_s$ rows and ζ columns. Frames are captured by a stationary camera so the background appears repeatedly in each frame except in extreme cases, i.e., the camera view is full of moving objects, which happens rarely in surveillance camera applications. Therefore, the background is typically

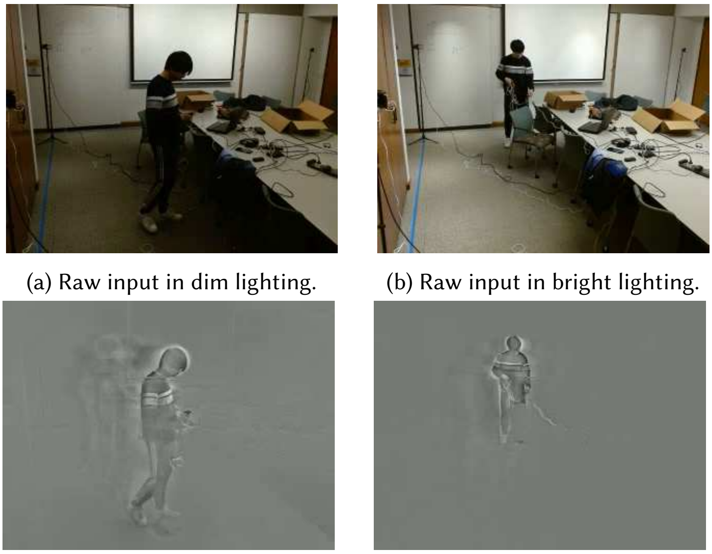
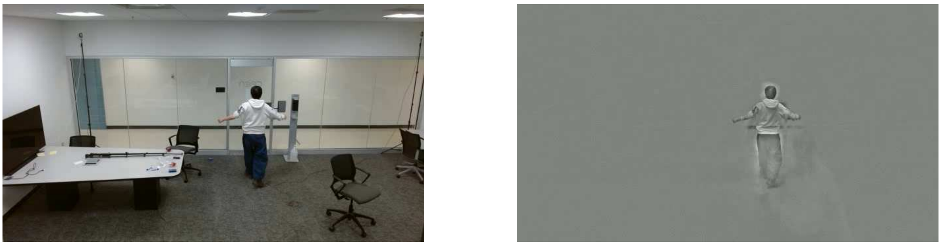

(a) Raw input in dim lighting.
(b) Raw input in bright lighting.
(c) Extracted foreground in dim lighting.
(d) Extracted foreground in bright lighting.

Fig. 3. Impact of lighting conditions on foreground (person) detection. SVD background removal is robust to variational lighting conditions.



(a) Original RGB image.

(b) Gray scale image after background removal.

Fig. 4. SVD background removal results.

the principal components of the matrix $\mathbf{A}$ and the background vectors form a low-rank matrix. The background is represented by the principal component of matrix $\mathbf{A}$, i.e., vectors with highest singular values.

SVD background removal consists of the following steps.

(1) Apply SVD on matrix $\mathbf{A}$, where $\mathbf{A}$ is defined as $\mathbf{UV}^\top$. In following steps, we follow Shlens's definitions [21].
(2) Select the top-k singular values and corresponding left and right singular vectors. The estimated background matrix $\mathbf{B} = \mathbf{U_{kk}V_k}^\top$ contains columns representing principal components of the video. $_\mathbf{k}$ is a diagonal matrix with top-k singular values on the diagonal. $\mathbf{V_k}$ indicates how $\mathbf{U_k}$ varies in magnitude over time.
(3) Foreground pixels are associated with moving objects. The foreground matrix is defined as $\mathbf{F} = \mathbf{A} - \mathbf{B}$. Each column in $\mathbf{F}_{norm}$ is a vectorized grayscale frame containing only the foreground.

Figure 4 shows the original frame and the frame after background removal.

3.2 Formulation and Objective Function

As shown in Figure 4, pixels belonging to the foreground and background are distinguished by how similar they are to gray. The background pixels are closer to gray, and the foreground pixels are farther (either brighter or darker than) gray. Based on this observation, we define the following concepts and expressions.

The value of a pixel p in frame $\mathbf{F}$ is $p_\mathbf{F} \in [0, 1]$. The mean pixel value for frame $\mathbf{F}$ is $\mu_\mathbf{F}$. Therefore, the deviation of $p_\mathbf{F}$ from the background defined as the *ungrayness*, is $\Gamma(p_\mathbf{F}) = |p_\mathbf{F} - \mu_\mathbf{F}|$.

The inclusion of background pixels in bounding boxes is penalized and the inclusion of foreground pixels is rewarded. A threshold is needed to distinguish background and foreground pixels according to their ungrayness. After subtracting this threshold from ungrayness, background pixels have negative scores and foreground pixels have positive scores. Given that background pixels are normally in the majority, foreground pixels may be viewed as outliers. Z-score-based outlier detection is used for determining a proper threshold. The Z-score is an efficient statistical measure that describes the position of a raw score in terms of its distance from the mean value, normalized to the standard deviation. A point is considered to be an outlier if its Z-score exceeds the preset threshold. Therefore, we define the *foreground score* of $p_\mathbf{F}$ as $\eta(p_\mathbf{F}) = \Gamma(p_\mathbf{F}) - \alpha \cdot \sigma_\mathbf{U}$, where α is an algorithmic parameter and $\sigma_\mathbf{U}$ is the standard deviation of pixel ungrayness. $\sigma_\mathbf{U}$ is initialized with the standard deviation of the first batch and incrementally updated with a weighted average of the current batch's standard deviation $\sigma_\mathbf{B}$ and the previous standard deviation. The weight is the number of frames N_B in the current batch that contain humans, divided by the batch size ζ, i.e., $\sigma_{\mathbf{U}} = \frac{N_B}{\zeta} \times \sigma_{\mathbf{B}} + (1 - \frac{N_B}{\zeta}) \times \sigma_{\mathbf{U}}$. Section 4.3 explains how to determine whether a human is in the scene using the UWB localization system. This updating algorithm is used, instead of using the standard deviation of the current batch, is due to the fact that when fading occurs for the entire batch, all pixels in the extracted foreground are close to gray, i.e., with small ungrayness. Simply using the standard deviation of the current batch, which is much smaller than the normal threshold, will cause background pixels to be misclassified as foreground pixels, making the fading compensation regions (see Section 4.4) too large.

In the following sections, pixels with positive foreground scores are called (privacy) *relevant* and pixels with negative foreground scores *irrelevant*. Relevant pixels are usually human-related. Irrelevant pixels are usually not human-related, except in the case of fading. To enable control over the importance of privacy in our application, a weight β is applied to relevant pixels. Therefore, the final foreground score, η, of a relevant pixel p is $\eta(p_{\mathbf{F}}) = \beta \cdot (\Gamma(p_{\mathbf{F}}) - \alpha \cdot \sigma_{\mathbf{U}})$. Given the number of bounding boxes, N, and the set of bounding boxes $\{B\}$, the cost function is

$$\mathcal{C}(\{B\}, \mathbf{F}) = -\sum_{p \in \cup\{B\}} \eta(p_{\mathbf{F}}) \text{ s.t. } |\{B\}| = N. \tag{1}$$

The optimized bounding boxe definition follows:

$$\{B^*\} = \underset{\{B\}}{\arg\min}\, \mathcal{C}(\{B\}, \mathbf{F}) \text{ s.t. } |\{B\}| = N. \tag{2}$$

3.3 Optimization Method

The optimization algorithm divides the bounding box estimate into two steps. It first specifies initial, potentially sub-optimal, bounding boxes. Next, the initial boxes are optimized based on the cost function described in Equation (1) using gradient descent. All bounding boxes in this article are represented as follows: $[(x_{\text{min}}, y_{\text{min}}), (x_{\text{max}}, y_{\text{max}})]$. A bounding box is defined by its upper left and lower right corners. Therefore, two coordinates are needed. In ERASE, the initial bounding boxes are generated using position information from the UWB **real-time localization system** (**RTLS**). Section 4.3 provides details. ERASE uses gradient descent to optimize the bounding boxes. Figure 5 shows the optimization algorithm.

3.4 Efficiency

Efficiency (measured by the frame rate achieved on inexpensive edge hardware in this article) is important for video analysis on edge devices because it influences the supported frame rate and cost of devices. We improve time efficiency in the following ways:

(1) downsampling to a lower resolution,
(2) using randomized SVD,
(3) adjusting the batch size and the number of top singular values for SVD,
(4) using appropriate halting conditions, and
(5) using an efficient method for gradient computation.

The frame resolution mainly affects computation time for matrix operations. There are many matrix operations in the system, e.g., SVD, matrix addition, and finding minimum and maximum elements. Matrix size increases with resolution and computation time increases with matrix size. Therefore, the system uses a downsampling strategy to improve efficiency. The video is first converted to grayscale and then downsampled to 160×120 for further processing. We discovered that this resolution is adequate for both SVD-based background removal and bounding box optimization.

The batch size and the number of top singular values used for SVD background removal significantly affect the computation time of SVD, especially on edge devices with constrained memory and computational power. Instead of computing full SVD, the system uses randomized SVD provided by Scikit-Learn [19], which efficiently computes a (usually very good) approximate truncated SVD. It is particularly fast on large matrices from which only a small number of components are required. Therefore, using fewer singular values speeds up SVD.

```
// Initialize bounding box set $\{B\}$, iteration count $n$.
Initialize $\{B\} \leftarrow \{B_{\text{init}}\}$, $n \leftarrow 1$, $improve \leftarrow \infty$
while $n \leq n_{\text{iter}}$ and $improve \geq \tau$ do
    $n \leftarrow n + 1$
    for all $B \in \{B\}$ do
        $(x_{\min}, y_{\min}, x_{\max}, y_{\max}) \leftarrow B$
        for all parameter $\theta \in \{x_{\min}, y_{\min}, x_{\max}, y_{\max}\}$ do
            Perturb box $B_{\text{temp}}$ by adding $\Delta_\theta$ to $\theta$
            Compute raw gradient:
                $\nabla_\theta C \leftarrow \frac{C(\{B_{\text{temp}}\}) - C(\{B\})}{\Delta_\theta}$
        end for
    end for
    Update $\{B\}$ using the computed gradient multiplied with
  the learning rate $lr$.
    Compute average improvement $improve$ (See Section 3.4).
end while
return $\{B\}$
```

Fig. 5. Bounding box optimization algorithm.

Setting an appropriate halting condition improves efficiency by reducing total optimization time with little impact on accuracy. At each iteration, we compute the average improvement of the cost over the previous five iterations. If the magnitude is below τ, the algorithm halts. τ is selected by manually analyzing the gradient descent process on more than 20 representative frames and choosing a value. These frames were not included in the testing data. According to our testing, the chosen τ based on the representative frames is robust and does not have to be adjusted for different environments.

We developed an efficient algorithm for gradient computation to accelerate the optimization process. Our initial approach computed the cost by masking the pixels that are in the bounding boxes and summing the foreground scores of these pixels. Each gradient computation per bounding box required eight such operations. However, we observed that frequent memory allocation for such large masking matrices incurred high overhead, which we reduced by redesigning the algorithm to avoid allocating new large matrices in each iteration. Instead of recomputing the entire cost, we calculate the difference region between the previous and the newly considered bounding box, slice the corresponding region from the foreground score matrix, and mask out any overlapping areas with other bounding boxes by setting those values to zero, thus ensuring that each pixel contributes to the cost only once. This sliced region, typically much smaller than the full matrix, is then summed to compute the cost. By operating only on the difference box and reusing the same memory locations, our method reduces the time required for each gradient descent step by at least 3×.

3.5 Fading Detection and Compensation

Background removal methods based on temporal analysis of scene changes such as MOG [22], **Geometric Median on Gaussian** (**GMG**) [25], and SVD, lack semantic understanding of the scene. They distinguish foreground from background based on object motion rather than the semantic features of the objects. If an object remains stationary, the deviations of related pixel intensities from the background decrease. We call this phenomenon *fading*. Fading causes disappearance of, or errors in, bounding boxes. During SVD background removal, if an object is stationary, related pixels become one of the principal components and are classified as background. Fundamentally, fading is the unfortunate result of using motion-based techniques to detect people when people are sometimes stationary. Figure 6 shows the first principal component (parts of the background) and the foreground in fading and non-fading conditions. When fading occurs, the human figure appears in the background and almost disappears in the foreground, which causes the bounding box to miss portions of the human. To solve this problem, we developed a method to detect and compensate for fading. Section 4.4 provides additional details.

4 Multi-Modal Human Detection and Tracking System

Using multi-modal measurements has the potential to improve the accuracy, efficiency, and robustness of vision-based human detection. ERASE fuses visual and UWB localization information. By projecting the UWB position estimates from physical coordinates to pixel coordinates, the system estimates the number of people in the camera view and initializes tentative bounding box solutions, thereby accelerating and improving the accuracy of vision-based human detection. The system then optimizes these bounding boxes using the cost defined in Section 3.2. The UWB-based location information also helps detect fading. This section describes our multi-modal fusion algorithm. Although the system is primarily motivated by privacy-preserving applications and it has a parameter tuned to this application (default object aspect ratio), the algorithm is somewhat robust to changes in this parameter, and the parameter can be tuned to other applications.

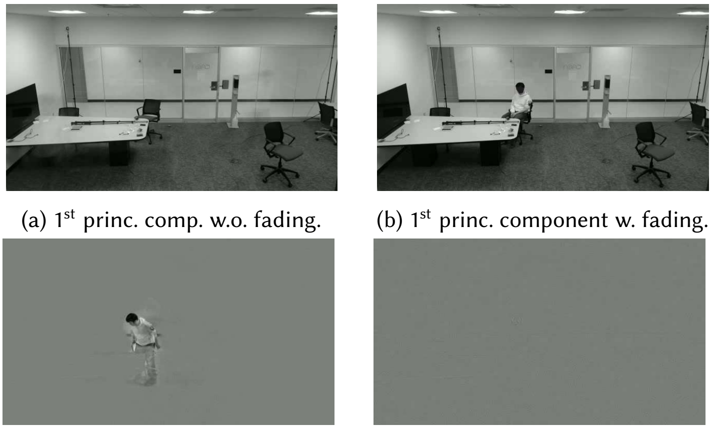

(a) 1st princ. comp. w.o. fading.
(b) 1st princ. component w. fading.
(c) Foreground without fading.
(d) Foreground with fading.

Fig. 6. First principal components and extracted foreground after SVD background removal in non-fading and fading conditions. When fading occurs, the human figure is in the principal components and disappears from the background, causing incomplete bounding box coverage.

4.1 Ultra-Wide Band Technology and Real-time Localization System

The localization system should be accurate, inexpensive, and easy to deploy. UWB-based localization systems are compatible with these requirements. UWB is a technology for transmitting data across a wide bandwidth (>500 MHz). This enables high signal power without interfering with conventional narrow-band and carrier wave transmissions in the same frequency bands. A UWB radio system can be used to determine transmission “time of flight” at various frequencies. This helps reduce multipath propagation errors because frequencies have different environment-dependent sensitivities to multipath effects. With a cooperative symmetric two-way metering technique, distances can be measured with high resolution and accuracy [1]. The average indoor localization error of a time-difference-of-arrival-based UWB RTLS is around 20 cm [23].

The UWB devices have wide and increasing availability; there are commercial chips, modules, and consumer products using UWB technology. For example, AirTag from Apple Inc. is popular due to its compactness, low cost, long battery life, and accurate localization results. It has the U1 UWB chip designed by Apple Inc. to enable very accurate short-range localization. UWB is also widely available in commercial smartphones, e.g., U1 in Apple smartphones and Exynos Connect

U100 in Samsung smartphones. We believe that UWB RTLS will be widely available for indoor venues such as malls and supermarkets in the near future.

We use MDEK1001 from Qorvo for real-time localization. It is a toolbox containing 12 nodes that can be configured as anchors or tags. It supports anchor auto-calibration, making implementation efficient and convenient. The tags periodically report their estimated positions to the server. Figure 7 shows an example of a position time-series from this localization system. In this example, a person holding a UWB tag walks along the route indicated by the red dashed line twice, counterclockwise.

UWB transceivers cost less than 15 USD, with four anchors being the minimum required for 3-D localization, i.e., the additional cost imposed by UWB localization is low. We expect that most people will have UWB transceivers available on their smartphones in the future, but it would also be possible for venue owners to provide tags to those entering the venue at a cost of less than 15 USD per tag. Calibrating the system takes less than 20 minutes.

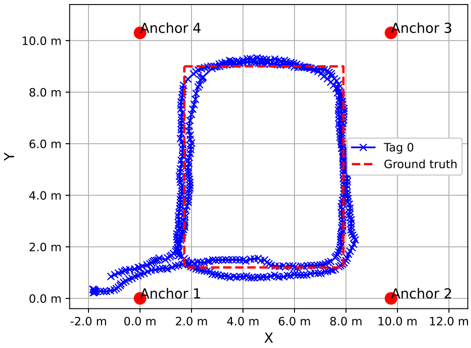


Fig. 7. Trace from the UWB-based RTLS. The UWB system can generally achieve sub-centimeter error.

4.2 Camera Calibration

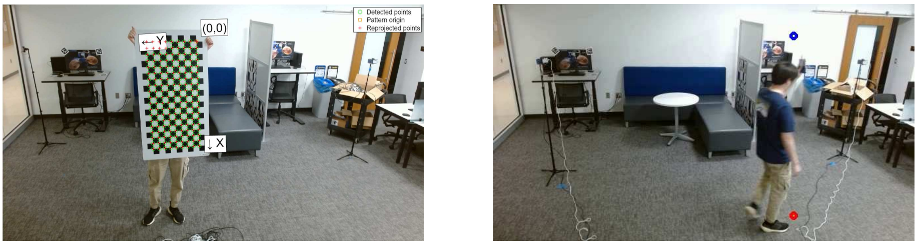
The localization system provides real-time locations of tags in 3-D real-world coordinates. The location is projected onto 2-D pixel coordinates for further processing. The MATLAB Camera Calibrator is used with checkerboard calibration [24] to obtain both camera intrinsic parameters and extrinsic parameters. Figure 8(a) shows an example image used for calibration. Based on the camera parameters, we form the intrinsic matrix **K** and the extrinsic matrix **R**.[1] Suppose the location of the tag in the real-world coordinates is $[x_w, y_w, z_w, 1]^\top$ (expressed as homogeneous coordinates). The projected location in pixel coordinates is $[x_w, y_w, z_w]^\top \cdot z_p = \mathbf{K} \cdot \mathbf{R} \cdot [x_w, y_w, z_w, 1]^\top$. Figure 8(b) provides an example of this projection. Given a point (x_w, y_w, z_w) in real-world coordinates, the red circle is the projection of $(x_w, y_w, 0\,\text{mm})$ and the blue circle is the projection of $(x_w, y_w, 2000\,\text{mm})$. We use 0 mm and 2,000 mm because they correspond to the ground and a position slightly above a typical person's head (note that the technique works for people of varying heights because the initial bounding box is later optimized). To summarize, the initial bounding boxes are over-sized and imprecise boxes that cover human figures. These initial bounding boxes are further optimized using visual information.

4.3 Bounding Box Proposal

To initialize bounding boxes in a frame, we need to know how many people are in the frame, their locations, and initial bounding box sizes.

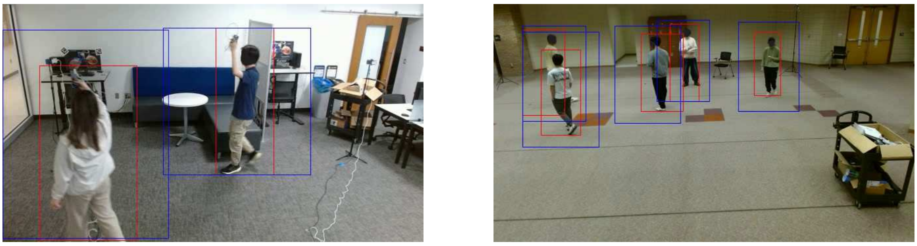
Given the resolution of a frame, (H_o, W_o), for a point in pixel coordinates (x_p, y_p), if $x_p \in [0, H_o) \land y_p \in [0, W_o)$, the point is within the frame. Assume that the position from the UWB

[1] The extrinsic parameters consist of a rotation, R, and a translation, t. It maps a point from physical coordinates to camera coordinates. The origin of the camera's coordinate system is at its optical center and its x-axes and y-axes define the image plane. The intrinsic parameters include the focal length, the optical center, also known as the principal point, and the skew coefficient.



(a) (b)

Fig. 8. (a) Example image for camera calibration. (b) Projected pixel (red dot) from $(x_w, y_w, 0\,\mathrm{mm})$ and projected pixel (blue dot) from $(x_w, y_w, 2000\,\mathrm{mm})$, in which the projected dots show that the UWB real-time localization system is able to adequately estimate the vertical span of the person for use in the initial bounding box proposal.



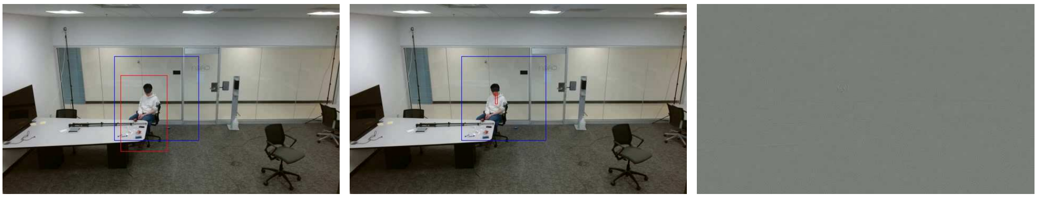
(a) (b)

Fig. 9. Examples of the UWB boxes (blue) as initial proposals and the final estimates (red) after optimization in (a) two-person cases and (b) five-person cases. The final estimates correctly converge to the human figures when different people are in the scene.

RTLS in real-world coordinates is (x_w, y_w, z_w). Since a person may partially appear in the scene (for example, the position of the tag after projection is not in the scene but the person's face is in the scene), we use two points, $(x_w, y_w, 0\,\mathrm{mm})$ and $(x_w, y_w, 2,000\,\mathrm{mm})$, to estimate the locations of the bottom and top of the person. We assume that the venue has a flat floor so that $z = 0\,\mathrm{mm}$ represents the ground plane and $z = 2,000\,\mathrm{mm}$ represents the approximate human head height plane. According to our tests, using this estimated height is robust and there is no need to adjust it based on venue or person. After projection, the locations in pixel coordinates are (x_{p1}, y_{p1}) (head) and (x_{p2}, y_{p2}) (foot). Arm spans are usually similar to heights. We therefore initialize a bounding box with height $h = y_{p1} - y_{p2}$ and width $w = h$. The bounding box is $[(x_{p1} - \frac{w}{2}, y_{p1}), (x_{p2} + \frac{w}{2}, y_{p2})]$. We call this box a *UWB box*. We consider a person to be in the scene if the UWB box intersects with the scene. The UWB boxes are approximate: Section 3.3 describes bounding box optimization. Figure 9 shows the initial UWB boxes and the final optimized bounding boxes in two-person and five-person cases. The UWB boxes provide the rough and potentially imprecise estimates of human figures and the final estimates successfully converge to the human figures.

4.4 Fading Detection and Compensation

Fading happens when an object ceases motion and appears to become a part of the background during SVD, which is common in real-world scenarios. For example, in a retail store, a customer may stop at a shelf to consider which product to buy. As explained in Section 3.5, a robust system should be able to detect and compensate for fading.



(a) With fading compensation. (b) Without fading compensation. (c) Foreground.

Fig. 10. The UWB box (blue) and final estimate (red) (a) with fading compensation and (b) without fading compensation. When fading happens, pixels in the foreground (c) are almost gray. Without fading compensation, as (b) shows, the final bounding box fails to cover the human figure exposing most privacy-relevant pixels. Fading compensation enables the corrects final bounding box shown in (a).

A literature search revealed only one study investigating fading in SVD-based background subtraction. Kajo et al. [9] describe an incremental tensor-based SVD method for abandoned object detection that decomposes video data into background and foreground components using spatiotemporal tensor slices and eigenfilter analysis. It is computationally efficient, suitable for real-time applications, requires no training data, and is robust to occlusion, illumination variation, and crowded environments, but it depends entirely on video data and empirical thresholds for detection.

In contrast, our approach avoids these dependencies by detecting fading conditions using location measurements. It is developed for a different scenario in which those subject to fading carry UWB tags (e.g., smartphones). This UWB RTLS informs on whether a user is stationary to identify fading. This approach introduces minimal computational overhead, requires only a single threshold parameter, and is efficient. Furthermore, it is robust to long-duration fading events, as it relies solely on the temporal consistency of UWB localization data. It does, however, require UWB infrastructure and can only detect stationary objects that are equipped with UWB tags.

In this subsection, we denote the location of a UWB tag at discrete time step t in real-world coordinates as (x_t, y_t, z_t). For fading detection, we only need horizontal coordinates (x_t, y_t). w_f is the window size for fading detection and ρ represents the threshold below which an object is considered stationary. A tag is considered stationary at time t if Equation (3) holds.

$$\bar{D}_t = \frac{1}{w_f} \sum_{\substack{k=-w_f/2 \\ k\neq 0}}^{w_f/2} \sqrt{(x_t - x_{t-k})^2 + (y_t - y_{t-k})^2} \leq \rho. \tag{3}$$

The historical bounding box right before fading happens, i.e., before motion ceases, is likely to correctly cover most of the person in the following frames. Considering that parts of the person may move out of this bounding box, e.g., raising arms, the final redaction box should merge the historical bounding box with information from the current frame. The system compensates for fading as follows. When fading is detected, a *historical estimate* bounding box associated with a tag is used. This estimate is updated every iteration in which fading is detected. The output of the gradient descent algorithm run on the current frame is called the *current estimate*. The final result is the union of the current estimate and historical estimate. Figure 10(a) and Figure 10(b) illustrate the final bounding box (red) with and without fading compensation. In Figure 10(b), the final estimate fails to cover the human figure due to fading and exposes most of the privacy-related pixels. The fading detection and compensation methods enable correct bounding boxes.

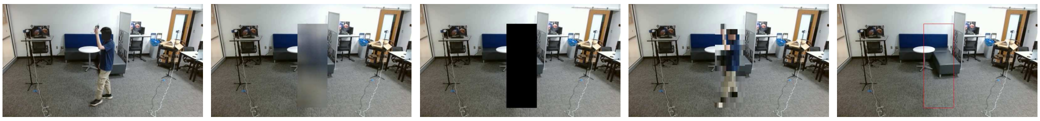

(a) Unredacted (b) Blur (c) Mask out (d) Pixelation (e) Replace

Fig. 11. The unredacted frame and the results in different redaction modes that the system supports.

4.5 Redaction Modes

After generating bounding box estimates, the system redacts pixels in bounding boxes. The system supports various redaction modes. Figure 11 shows the unredacted and redacted frames using different redaction modes including blurring, masking, pixelation, and invisibility (i.e., replacing pixels inside bounding boxes with the pre-captured background image). Users can select the mode based on application needs. For example, there may be scenarios in which pose estimation is required and is feasible after applying blurring or pixelation, but masking the bounding boxes would prevent pose estimation. Discussions of the impact of different privacy redaction methods on vision-based deep learning models are out of the scope of this article, but appear in other works [5, 10, 18].

5 Evaluation

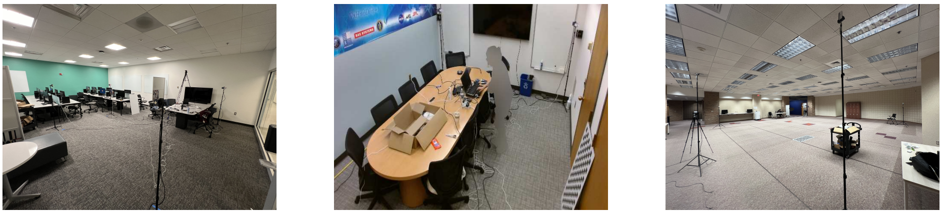
This section describes our experimental setup, evaluation dataset, algorithmic parameters, and explains the implications of our experimental results.

5.1 Hardware Setup

As mentioned previously, ERASE is designed for implementation on inexpensive, commodity edge devices. This subsection indicates the hardware we use and the costs of our system. The system uses the Qorvo MDEK1001, which includes 12 packaged UWB nodes with pre-loaded firmware. It costs 299 USD for the UWB RTLS and 25 USD per node. Note that we use this expensive toolbox because it provides visualization services useful during experimentation. Venue owners might further reduce the cost by using less expensive UWB nodes (around 15 USD). Four anchors are required to build a localization system for a venue at a cost of around 60 USD. Many smartphones already contain UWB transceivers (tags), e.g., iPhones since iPhone 11. We believe this will be more common in the future, making it available in most scenarios. It would also be possible for the venue owners to provide tags to those entering the venue at a cost of less than 15 USD per tag.

The requirements of our target application scenarios impose constraints on the computational hardware used for evaluation. First, it must be cost-effective (≤ 50 USD per unit) to facilitate large-scale deployment. This generally implies that it uses a CPU common in IoT edge devices, such as those in the Cortex-A series. Finally, the board should provide sufficient memory (at least around 2 GB) to support efficient data processing, especially matrix operations. To facilitate development and expedite system verification, we also consider the availability of well-supported drivers for commonly used peripheral devices, such as cameras. Based on these considerations, the ROCKPro64 single-board computer was an option for experimental evaluation. Although the Raspberry Pi 4B has similar performance at a lower cost, the ROCKPro64 was already available in our lab and was therefore used for experimentation.

ERASE is evaluated on a ROCKPro64 single-board computer costing 80 USD with a Rockchip RK3399 hexa-core processor (dual-core 1.8 GHz ARM Cortex A72 and quad-core 1.4 GHz ARM Cortex A53), and 4 GB of LPDDR4 system memory. The ROCKPro64 consumes 5–8 watts under CPU-intensive loads. The ARM Cortex A72 is a commonly used CPU on IoT development boards,



(a) Computer lab (b) Conference room (c) Library

Fig. 12. Data were gathered in three indoor spaces with different room sizes, furniture layouts, and lighting conditions to evaluate the system's sensitivity to these variations.

e.g., the Raspberry Pi 4B. We used the ROCKPro64 for evaluation, but intentionally avoided using GPU acceleration to make its performance very similar to less expensive development boards using the same processor, such as the 35 USD Raspberry Pi 4B.

The system implementation is inexpensive and can be rapidly deployed. It generally takes us 10 minutes to calibrate a camera with MATLAB Camera Calibrator and 20 minutes to calibrate the UWB subsystem with 4 anchors.

5.2 Dataset

Evaluation of ERASE requires a UWB/vision time-synchronized dataset. There is an existing multi-modal dataset [3] containing video, human traces from the UWB RTLS, and acceleration from accelerometers. However, in this dataset, data are collected for only one person at a time. We believe it is important to evaluate our system with a varying number of people in the scene. Therefore, we collected a dataset which we plan to make available for use by other researchers. The dataset contains time-synchronized video data and UWB position data.

To evaluate ERASE in various indoor scenarios, we collected data from three indoor venues with different furniture layouts, room sizes, and lighting conditions. Table 1 summarizes the dimensions and lighting intensity ranges of each venue. To measure lighting intensities, we used a lux meter at approximately 20 evenly distributed locations within each venue and report the ranges of the lux values. Note that none of the venues is extremely dark because public venues such as grocery stores and hospitals are typically designed with adequate light for human vision. Figure 12 shows the layouts.

Table 1. Venue Parameters

Venue	**Dimension (m)**	**Lighting conditions (lux)**
Conference Room	4.8 × 3.5	492–783
Computer Lab	9.5 × 6.9	305–479
Library	10.5 × 11	491–699

Participants hold UWB tags in their hands at waist level. Non-line-of-sight localization errors tend to be higher when tags are carried in pockets, and decrease when held in the hands. Previous work [17] discusses the impact of sensor positions on human bodies on UWB ranging. There is work [4, 14] on detecting and mitigating errors in non-line-of-sight conditions. With compensation for non-line-of-sight, the UWB system is able to accurately estimate user positions even when UWB tags are carried in pockets. We also vary the number of people in the scene from one to five to measure the impact on accuracy and time-efficiency. In total, the dataset contains 60 minutes of video at 10 FPS (36,000 frames) and time-synchronized human traces from the UWB RTLS at the same frequency.

We provide ground truth bounding boxes for each video. We do so by running YOLO11x, the most powerful model in the YOLO11 series, on the videos to produce estimated bounding boxes, inspecting all frames with predicted bounding boxes, and manually drawing corrected bounding boxes on frames for which YOLO11x erred.

5.3 Algorithmic Parameters

One of our design goals is to minimize the number of algorithmic control parameters requiring tuning by users. This section defines all the algorithmic parameters and indicates how they were determined. The system has the following parameters: ζ, k, α, β, τ, w_f, ρ, lr, and $niter$. Table 2 indicates the values of these parameters. We hold each parameter constant over all locations, i.e., we do not tune the parameters to specific locations.

Table 2. Algorithmic Parameters

Parameter	Value	Name
ζ	50	SVD batch size
k	5	SVD top-k components
α	1	score threshold weight
β	3	foreground score multiplier
τ	0.5	improvement threshold for halting
w_f	10	fading detection window size
ρ(m)	0.15	fading detection threshold
lr	1	learning rate
$niter$	20	maximum iterations

ζ is the SVD batch size. If ζ is too small, it will violate the low-rank assumption of SVD background removal because the background vector is less principal and representative of the batch. If, that is, too large, it increases memory requirements and computation time. k determines the number of singular values and related singular vectors representing the background. As k increases, more singular vectors are classified as the background.

As shown in the definition of the pixel foreground score in Section 3.2, α can be used to control the threshold to determine relevant and irrelevant pixels. Increasing α reduces the number of relevant pixels. Therefore, by selecting a proper α, the system is able to correctly assign foreground scores to pixels during gradient descent, and bounding boxes are able to converge to human figures. Reducing α reduces the probability of misidentifying relevant pixels as irrelevant, but increases the probability of misidentifying irrelevant pixels as relevant. For most privacy-relevant applications, using a smaller α is appropriate because leaking a relevant pixel is worse than redacting an irrelevant pixel.

β is a weight given to foreground scores of relevant pixels. β weights foreground scores of relevant pixels (privacy-related foreground) and irrelevant pixels (privacy-unrelated background) differently during optimization. If a system assigns the same weights to relevant and irrelevant pixels, some privacy-relevant pixels may escape redaction, for example, people's faces might be exposed. We therefore introduce the weighting parameter β allowing trade-offs between false negatives and false positives. The more important privacy is relative to mistaken redaction of privacy-irrelevant pixels, the larger β should be.

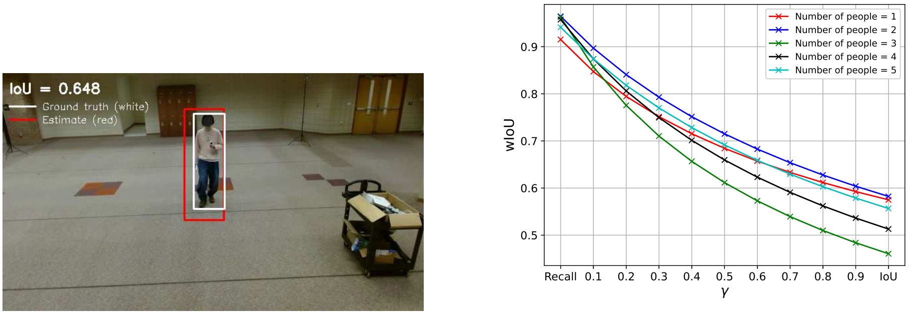
τ is a threshold defined in Section 3.4. It determines the halting condition for gradient descent.

w_f is the window size in fading detection of ERASE (see Section 4.4). As w_f decreases, fading detection becomes more sensitive. ρ is the distance threshold in meters for fading detection in ERASE. Within a window, if the average distance between previous positions and the current position in real-world coordinates is below ρ, it is detected as fading.

lr is the learning rate in gradient descent. $niter$ is the maximum number of iterations.

5.4 Measures

Results for three privacy-specific measures are provided. We report redaction success rates for three regions of interest: the whole body, the upper body, and the face. We visually check whether any



(a) (b)

Fig. 13. (a) The estimate (red) successfully covers all privacy-relevant pixels but includes some privacy-irrelevant pixels; it would be adequate for many privacy protection applications. However, the IoU of the ground truth (white) and the estimate is only 0.648, which is low. In contrast, weighted IoU can be used to penalize leakage of privacy-relevant pixels more heavily than inappropriate redaction of privacy-irrelevant pixels. (b) Increasing γ (see Section 5.4), penalizes the redaction of privacy-irrelevant pixels more heavily. The best value of γ is application-dependent. With $\gamma = 0$ wIoU is equal to recall. With $\gamma = 1$, wIoU is equal to IoU. The main error source for ERASE is redaction of privacy-irrelevant pixels.

region of interest is exposed after redaction in each frame. These measures are useful because they are directly privacy-related, but they do not consider how much useful background information is maintained. In other words, one might achieve a 100% success rate by redacting the whole scene, i.e., transmitting nothing. Therefore, we propose other privacy measures.

Traditional **intersection over union** (**IoU**) has been used to evaluate the quality of bounding box estimates. It is defined as $\frac{TP}{TP+FP+FN}$. TP is true positive, FN is false negative, and FP is false positive. However, IoU is inappropriate for privacy-preserving applications because false positives and false negatives should often have different penalties. From the perspective of privacy, false negatives imply the leakage of private information and false positives imply unnecessary redaction of privacy-irrelevant data. Leaking a pixel is generally much more harmful than inappropriately redacting a pixel. As shown in Figure 13(a), the estimate (red) successfully covers all privacy-relevant pixels while including a few privacy-irrelevant pixels, which is an acceptable estimate for privacy-relevant applications. However, the IoU of the ground truth (white) and the estimate is only 0.648, a poor score for IoU.

Motivated by the problem of IoU and inspired by the definition [2] of weighted IoU for segmentation, we define a measure, also called wIoU, allowing different weights for false negatives and false positives for privacy-relevant applications. wIoU is defined as $\frac{TP}{TP+\gamma\times FP+FN}$. $\gamma \in [0, 1]$ is an algorithmic parameter allowing adjustment of the penalty. In scenarios where privacy is more important and privacy-irrelevant information is less important, a lower γ should be used. With $\gamma = 1$, wIoU is equal to IoU. With $\gamma = 0$, wIoU is equal to recall, defined as $\frac{TP}{TP+FN}$. Recall is useful for determining the proportion of privacy-relevant pixels that are covered by the estimated bounding boxes, but it does not penalize inclusion of privacy-irrelevant pixels in bounding boxes, which means a method can simply maximize recall by redacting the whole scene, i.e., transmitting nothing. IoU penalizes the inclusion of privacy-irrelevant pixels too much relative to missing privacy-relevant pixels. With $\gamma = 0.3$, the wIoU of boxes in Figure 13(a) is 0.849. In Figure 13(b), by sweeping γ from 0 to 1, the measure penalizes the inclusion of privacy-irrelevant pixels more heavily. The best value of γ is application-dependent. The error of our system mainly comes from incorrectly classifying privacy-irrelevant pixels as privacy-relevant, rather than exposure of privacy-relevant pixels. In this article, we use wIoU with $\gamma = 0.3$ (i.e., $wIoU_{0.3}$) for evaluation.

5.5 Efficiency and Accuracy Comparison

We evaluate the efficiency and accuracy of ERASE and compare it with other approaches: YOLO11m [8], YOLO11n (the YOLO11 model with the fewest parameters), MediaPipe [12]'s pose model, and MediaPipe's object detection model.[2] YOLO11m has 20.1 M parameters and YOLO11n is a tiny model with 2.6 M parameters. MediaPipe's models have 3.37 M parameters. Evaluation is conducted when different numbers of people are in the scene.

As Figure 14 shows, the time efficiency of ERASE is approximately linearly related to the number of people because the number of iterations is linearly related to the number of bounding boxes for each frame. When one person is in the scene, the system runs at over 60 FPS, and when five people are in the scene, it runs at over 20 FPS. We believe 10 FPS is the lowest acceptable rate for most applications requiring video streaming of human activities. However, YOLO11n runs at less than 4 FPS on the ROCKPro64: too slow for real-time use. ERASE is five times as fast as YOLO11n.

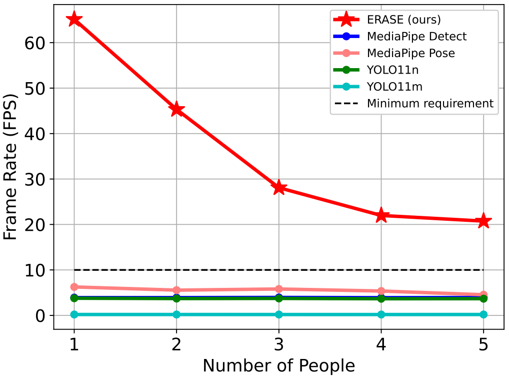


Fig. 14. Note that the lines for YOLO11n and MediaPipe Detect overlap. ERASE is able to run at over 20 FPS, approximately 4× faster than the neural network models, when five people are in the scene.

Figure 15 compares the accuracy of YOLO11m, YOLO11n, ERASE (ours), MediaPipe detect, and MediaPipe pose, in recall, $wIoU_{0.3}$, IoU, and precision, when different numbers of people are in the scene. Recall indicates the proportion of privacy-relevant pixels redacted. The precision, defined as $\frac{TP}{TP+FP}$, is the proportion of privacy-relevant pixels in the estimated bounding boxes. Inclusion of too many privacy-irrelevant pixels results in a very low precision. YOLO11m achieves the best accuracy in all measures, but is too slow for real-time use on inexpensive, commodity IoT hardware. ERASE achieves good results in recall and $wIoU_{0.3}$, i.e., it redacts most privacy-relevant pixels while maintaining useful environment information. However, it has lower precision compared to other neural network models, implying that its bounding boxes include more privacy-irrelevant pixels than others.

We found that for the dataset collected in the smallest room shown in Figure 12(b), where the person is close to the camera, none of the tested neural network models had a high recall. This is likely due to the human figures being closer (and therefore appearing larger than typical in the training data) and visual occlusion by furniture. However, ERASE had consistent accuracy because SVD background removal is scale-insensitive. ERASE exhibits robustness to complex human poses. As shown in Figure 16, the predicted bounding boxes (in red) accurately capture human figures, even under partial occlusion and across a diverse range of poses, such as sitting and bending. Leveraging the nature of SVD background subtraction, which detects motion rather than relying on semantic or visual features, ERASE detects the human figures, regardless of pose complexity and occlusion conditions.

[2] For YOLO models, results are computed with $conf = 0.5$ and $imgsz = 320$. For MediaPipe's pose, results are computed with $num\_poses = 5$; $min\_pose\_detection\_confidence = 0.35$; $min\_pose\_presence\_confidence = 0.35$. For MediaPipe's object detection, results are computed with $category\_allowlist =$ ["person"]; $max\_results = 5$; $score\_threshold = 0.35$.

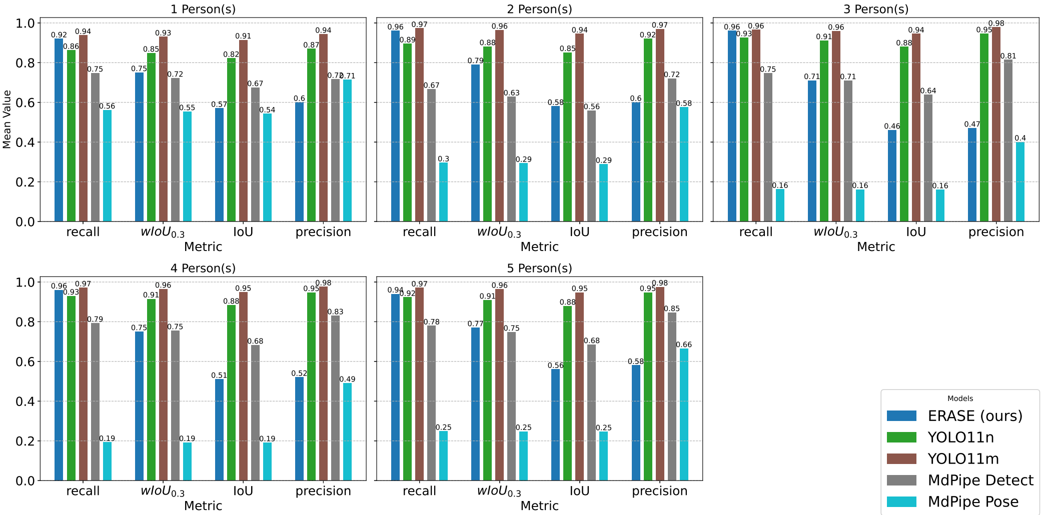
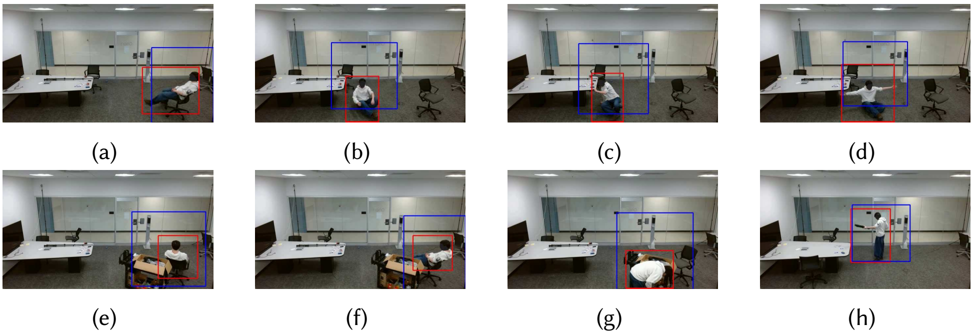


Fig. 15. Accuracies as functions of redaction method and number of people in scenes. YOLO11m achieves the best accuracy in all measures. ERASE achieves good results in recall and $wIoU_{0.3}$, which implies that it is able to redact most privacy-relevant pixels without redacting many privacy-irrelevant pixels. However, it has lower precision compared to the neural network models, implying that its estimated boxes cover more privacy-irrelevant pixels. ERASE is the only one capable of running in real-time.


(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 16. ERASE is capable of detecting individuals across a variety of poses and under partial occlusion. The predicted bounding boxes (in red) accurately enclose the human figures in these challenging scenarios.

Figure 17 shows the accuracies of redaction for heads, upper bodies, and whole bodies. Ground truth is determined by manual inspection of every frame. The strictest possible standard is used: if even a pixel of the relevant body region is exposed, the redaction is considered a failure. For example, if four people are successfully redacted and a pixel of the ear of the last person is exposed, this is considered a failure for the frame, hence the decrease in accuracy when more people are present. Most failures in multi-person frames only affect one person. ERASE has a more than 90% success rate in redacting human heads (faces) and more than 77% success rate in redacting human upper bodies. It has a relatively lower whole body redaction success rate, especially for the five-person case. Most failures result from exposure of the lower legs. Compared to the torso, the human lower legs cover a smaller area in the scene, resulting in more irrelevant pixels and fewer relevant pixels near the bottom of the bounding boxes. During optimization, this may cause the region to be excluded from the bounding box.

5.6 Limitations and Caveats

There are several limitations to ERASE. It requires UWB RTLS deployment, although this is inexpensive (less than 15 USD per node) and rapidly deployable (20 minutes to calibrate four anchors). It is designed for venues in which the owners have the right to install cameras and UWB infrastructure, i.e., we do not focus on venues without owners. It also requires that customers have smartphones or other mobile devices that support UWB localization, or that venue owners provide UWB tags to customers, which would introduce cost and maintenance effort. Although ERASE is robust to some UWB RTLS error, extreme errors resulting from signal occlusion that reduces the number of usable anchors below four would reduce redaction accuracy, perhaps catastrophically. In this situation, reducing such non-line-of-sight errors, e.g., by adding additional UWB anchors, is likely the most straightforward way to improve system accuracy. Lack of semantic understanding is another limitation of ERASE. It cannot distinguish between humans and other moving objects that may appear in the venue, e.g., dogs or robots. This is a (possibly necessary) tradeoff to enable use on performance-constrained commodity edge hardware. As GPU hardware costs decrease, it may become feasible to deploy deep neural networks such as YOLO on edge devices at scale. At present, the high computational and financial costs associated with such deep neural networks remain significant barriers to deployment.

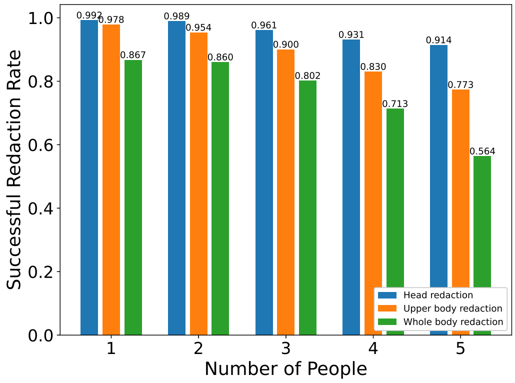


Fig. 17. Success rates for redacting heads, upper bodies, and whole bodies for various numbers of people in the field of view. After redaction by ERASE, most privacy-relevant pixels are successfully removed.

6 Conclusion

This article has described ERASE, an efficient real-time multi-modal human tracking system capable of running on low-end edge devices. Our goal is to enable inexpensive on-device video redaction for privacy protection. We tested ERASE on a dataset collected in three representative indoor locations using only inexpensive, commodity edge devices. The multi-modal system operates at 20–60 FPS, achieves a $wIoU_{0.3}$ score of 0.71–0.79, and succeeds in completely redacting all privacy-sensitive pixels at a rate of 91%–99% in human head regions and 77%–91% in upper body regions, varying with the number of people present in the field of view. ERASE enables accuracy and efficiency for real-time privacy-preserving visual human tracking on inexpensive, commodity edge devices. It is robust to changing lighting conditions, as well as variations in human scale, distance, and pose, without requiring retraining or parameter tuning. Our results demonstrate the practicality of real-time edge redaction in human tracking applications using inexpensive, commodity hardware.

Acknowledgment

The authors gratefully acknowledge the contributions of A. Thomare, B. Bolthouse, B. Susanto, P. Yim, R. Guan, S. Akhmatdinov, Yanbo Wang, Yu Wang, and Y. Zhong for their efforts in test field setup and data collection. Special thanks to S. Zhou for her help manually checking redacted video frames to determine the rate of private information leakage. Their assistance was indispensable to this work.

References

[1] Michal Aftanas, Jana Rovnáková, Milos Drutarovsky, and Dusan Kocur. 2008. Efficient method of TOA estimation for through wall imaging by UWB radar. In *Proceedings of the IEEE International Conference on Ultra-Wideband*, Vol. 2. IEEE, 101–104.

[2] Yeong-Jun Cho. 2024. Weighted Intersection over Union (wIoU) for evaluating image segmentation. *Pattern Recognition Letters* 185 (2024), 101–107. DOI: https://doi.org/10.1016/j.patrec.2024.07.011

[3] Mathias Ciliberto, Vitor Fortes Rey, Alberto Calatroni, Paul Lukowicz, and Daniel Roggen. 2021. Opportunity ++: A Multimodal Dataset for Video- and Wearable, Object and Ambient Sensors-based Human Activity Recognition. DOI: https://doi.org/10.21227/vd6r-db31

[4] Mengyao Dong. 2021. A low-cost nlos identification and mitigation method for UWB ranging in static and dynamic environments. *IEEE Communications Letters* 25, 7 (July 2021), 2420–2424. DOI: https://doi.org/10.1109/LCOMM.2021.3070311

[5] Håkon Hukkelås and Frank Lindseth. 2023. Does image anonymization impact computer vision training?. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 140–150.

[6] Yasha Iravantchi, Thomas Krolikowski, William Wang, Kang G. Shin, and Alanson Sample. 2024. PrivacyLens: On-device PII removal from RGB images using thermally-enhanced sensing. *Proceedings on Privacy Enhancing Technologies* 2024, 4 (July 2024), 872–891. DOI: https://doi.org/10.56553/popets-2024-0146

[7] Matthew Ishige, Yasuhiro Yoshimura, and Ryo Yonetani. 2024. Opt-in camera: Person identification in video via UWB localization and its application to opt-in systems. arXiv:2409.19891. Retrieved from https://arxiv.org/abs/2409.19891 (2024).

[8] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. 2023. *Ultralytics YOLO.*

[9] Ibrahim Kajo, Nidal Kamel, and Yassine Ruichek. 2018. Incremental tensor-based completion method for detection of stationary foreground objects. *IEEE Transactions on Circuits and Systems for Video Technology* 29, 5 (May 2018), 1325–1338.

[10] Jun Ha Lee and Su Jeong You. 2024. Balancing privacy and accuracy: Exploring the impact of data anonymization on deep learning models in computer vision. *IEEE Access* 12 (2024), 8346–8358. DOI: https://doi.org/10.1109/ACCESS.2024.3352146

[11] Xin Liu, Guoying Zhao, Jiawen Yao, and Chun Qi. 2015. Background subtraction based on low-rank and structured sparse decomposition. *IEEE Transactions on Image Processing* 24, 8 (April 2015), 2502–2514. DOI: https://doi.org/10.1109/TIP.2015.2419084

[12] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, and Matthias Grundmann. 2019. MediaPipe: A framework for perceiving and processing reality. In *Proceedings of the Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition.* Retrieved from https://mixedreality.cs.cornell.edu/s/NewTitle_May1_MediaPipe_CVPR_CV4ARVR_Workshop_2019.pdf

[13] Huizhen Mu, Chao Yu, Shuna Jiang, Yujing Luo, Kun Zhao, and Wen Chen. 2024. Indoor pedestrian positioning method based on ultra-wideband with a graph convolutional network and visual fusion. *Sensors* 24, 20 (October 2024), 6732.

[14] Quoc-Tuong Ngo, Pierre Roussel, Bruce Denby, and Gerard Dreyfus. 2015. Correcting non-line-of-sight path length estimation for ultra-wideband indoor localization. In *Proceedings of the International Conference on Localization and GNSS.* 1–6. https://doi.org/10.1109/ICL-GNSS.2015.7217140

[15] Thien Hoang Nguyen, Thien-Minh Nguyen, and Lihua Xie. 2021. Range-focused fusion of camera-IMU-UWB for accurate and drift-reduced localization. *IEEE Robotics and Automation Letters* 6, 2 (April 2021), 1678–1685.

[16] Nuria M Oliver, Barbara Rosario, and Alex P Pentland. 2000. A bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 8 (August 2000), 831–843.

[17] Timothy Otim, Alfonso Bahillo, Luis Enrique Díez, Peio Lopez-Iturri, and Francisco Falcone. 2019. Impact of body wearable sensor positions on UWB ranging. *IEEE Sensors Journal* 19, 23 (August 2019), 11449–11457. DOI: https://doi.org/10.1109/JSEN.2019.2935634

[18] José Ramón Padilla-López, Alexandros Andre Chaaraoui, and Francisco Flórez-Revuelta. 2015. Visual privacy protection methods: A survey. *Expert Systems with Applications* 42, 9 (June 2015), 4177–4195.

[19] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* 12, null (2011), 2825–2830.

[20] Pingping Peng, Chao Yu, Qihao Xia, Zhengqi Zheng, Kun Zhao, and Wen Chen. 2022. An indoor positioning method based on UWB and visual fusion. *Sensors* 22, 4 (Feburary 2022), 1394.

[21] Jonathon Shlens. 2014. A Tutorial on principal component analysis. Retrieved from https://arxiv.org/abs/1404.1100

[22] C. Stauffer and W.E.L. Grimson. 1999. Adaptive background mixture models for real-time tracking. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2. IEEE, 246–252. DOI: https://doi.org/10.1109/CVPR.1999.784637

[23] Dadiao Tian and Qiang Xiang. 2020. Research on indoor positioning system based on UWB technology. In *Proceedings of the IEEE Information Technology and Mechatronics Engineering Conference*. IEEE, 662–665.

[24] Zhengyou Zhang. 2002. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 11 (August 2002), 1330–1334.

[25] Z. Zivkovic. 2004. Improved adaptive gaussian mixture model for background subtraction. In *Proceedings of the International Conference on Pattern Recognition*, Vol. 2. 28–31. DOI: https://doi.org/10.1109/ICPR.2004.1333992

Received 13 August 2025; revised 13 August 2025; accepted 13 August 2025