Publishing Video Data with Indistinguishable Objects

Han Wang
Illinois Institute of Technology
hwang185@hawk.iit.edu

Yuan Hong
Illinois Institute of Technology
yuan.hong@iit.edu

Yu Kong
Rochester Institute of Technology
yu.kong@rit.edu

Jaideep Vaidya
Rutgers University
jsvaidya@rbs.rutgers.edu

ABSTRACT

Millions of videos are ubiquitously generated and shared everyday. Releasing videos would be greatly beneficial to social interactions and the community but may result in severe privacy concerns. To the best of our knowledge, most of the existing privacy preserving techniques for video data focus on detecting and blurring the sensitive regions in the video. Such simple privacy models have two major limitations: (1) they cannot quantify and bound the privacy risks, and (2) they cannot address the inferences drawn from the background knowledge on the involved objects in the videos. In this paper, we first define a novel privacy notion *ϵ-Object Indistinguishability* for all the predefined sensitive objects (e.g., humans and vehicles) in the video, and then propose a video sanitization technique VERRO that randomly generates utility-driven synthetic videos with indistinguishable objects. Therefore, all the objects can be well protected in the generated utility-driven synthetic videos which can be disclosed to any untrusted video recipient. We have conducted extensive experiments on three real videos captured for pedestrians on the streets. The experimental results demonstrate that the generated synthetic videos lie close to the original video for retaining good utility while ensuring rigorous privacy guarantee.

1 INTRODUCTION

Millions of videos are ubiquitously generated and shared everyday via video surveillance devices, traffic cameras, smart phones, among others. Sharing such video data would greatly benefit human interactions and the community. For instance, surveillance cameras in buildings capture possible threats to the corporate assets (such videos are shared for analysis in many cases [44]). Traffic videos contribute to monitoring the street traffic and traffic analysis applications such as vehicle counting and traffic flow analysis [3] as well as pedestrian behavior analysis.

However, these scenarios have often raised severe privacy concerns since human faces, bodies, identities, activities and other sensitive information can be recorded in such videos [8, 44]. Thousands of vehicles are involved in a traffic monitoring video, and drivers may not be willing to share their vehicle plate, make, model, locations and trajectories [2]. In addition, video surveillance systems monitor specific areas of interests with the following goals: law enforcement, personal safety, resource planning, and security of assets [44]. While ensuring safety and deterrence, it may also compromise the privacy of innocent individuals. Thus, privacy preserving solutions for videos have attracted significant interests recently.

To the best of our knowledge, most of the existing privacy preserving techniques for addressing the privacy concerns in video data (e.g., [1, 6, 20, 24, 42]) focus on detecting and blurring the sensitive regions in the videos (e.g., faces and bodies). Such simple privacy models have two major limitations: (1) they cannot quantify the bound of privacy risks, and (2) they cannot address the inferences drawn from the background knowledge on the involved objects. For instance, the video recipient may have known that an individual lives near the scene, and he/she usually wears red clothes as well as likes running at a specific side of the street. In this case, even if all the humans can be detected and blurred before video disclosure, they can be readily re-identified by the adversary with the above background knowledge.

ϵ-Object Indistiguishability. To tackle such critical limitations, we define a novel privacy notion for protecting the objects (and the corresponding individuals) in the videos – "*ϵ-Object Indistinguishability*", which is extended from the emerging *differential privacy* in local setting [4, 10, 16]. Specifically, in the past decade, the notion of differential privacy has emerged essentially as the de facto privacy standard for bounding the privacy risks while sanitizing different data [7, 13, 25]. Adversaries cannot infer if a certain individual is included in the input or not from the noisy aggregated result (perturbed by a trusted aggregator) regardless of their background knowledge [13]. More recently, local differential privacy (LDP) models have been proposed to privately perturb data by each individual such that the collected (random) data from different individuals can be *indistinguishable.* Inspired by the LDP models, our privacy notion also ensures *indistinguishability* for all the objects in the randomized output video, and thus the perturbed video can be safe to be disclosed to any untrusted video recipient.

Recall that videos differ from many other data (e.g., statistical databases [13], location data [32], and search logs [26]). A local video may include numerous objects corresponding to multiple different individuals, e.g., many pedestrians are recorded in a single video, and many vehicles are recorded in the same video. A video includes the "local data" of many individuals (e.g., humans) which will be shared to the untrusted recipients via the video owner. Thus, the primary difference between *ϵ-Object Indistinguishability* and the original definition of LDP [4, 10, 16] is that the video owner locally perturbs data for all the objects rather than letting the objects execute perturbation (see Figure 1).

Contributions. With the privacy notion of *ϵ-Object Indistinguishability*, we propose a video sanitization technique that randomly generates a synthetic video by the video owner (e.g., the agency which captures the video) while ensuring *ϵ-Object Indistinguishability* and good utility.

Specifically, we design a novel random response scheme (by optimizing the RAPPOR [16]) that randomly generates different objects in the video by maximizing the utility of random response

Published in Proceedings of the 23rd International Conference on Extending Database Technology (EDBT), March 30-April 2, 2020, ISBN 978-3-89318-083-7 on OpenProceedings.org.

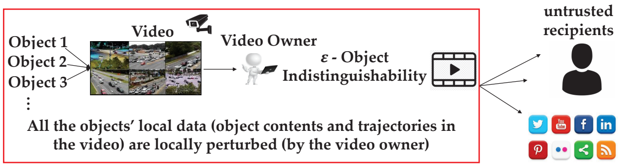

Video
Video Owner
Object 1
Object 2
Object 3
⋮
ε - Object Indistinguishability
All the objects' local data (object contents and trajectories in the video) are locally perturbed (by the video owner)
untrusted recipients

Figure 1: VERRO: Ensuring *Object Indistinguishability* in the Video Data Sanitization

[16] applied to the objects. Thus, we name our proposed technique as *"Video with Randomly Responded Objects* (VERRO)". As a result, we boost the utility of VERRO in two folds: (1) for each object, optimizing its random response in different frames, and (2) interpolating the trajectories of objects in the video [17] (without additional privacy leakage [14], see Section 4). Thus, the synthetic video can be disclosed to any untrusted recipient. Finally, we summarize the major contributions of this paper as below:

- To the best of our knowledge, we define the first rigorous privacy notion for all the sensitive objects (predefined by the video owner) in the video data, which ensures that all the objects are *indistinguishable* in the randomized output video against arbitrary background knowledge.
- We propose a novel video sanitization technique VERRO that randomly generates utility-driven synthetic videos in which any two sensitive objects are *ϵ-Object Indistinguishable.*[1] The video owner can also specify its privacy budget ϵ for all the objects in its video.
- The proposed novel random response scheme (by optimizing RAPPOR [16]) in VERRO that optimally picks frames of the video to randomly generate objects (while satisfying indistinguishability). The utility of the synthetic video is further improved using computer vision techniques.
- We have conducted extensive experiments on real videos to validate the performance of VERRO. The experimental results demonstrate that VERRO can effectively generate private synthetic videos with high utility.

The remainder of this paper is organized as follows. Section 2 introduces some preliminaries. Section 3 illustrates the first phase of VERRO and analyzes the privacy guarantee. Section 4 presents the second phase of VERRO (for further boosting the utility) and its privacy guarantee. Section 5 gives discussions for VERRO. Section 6 demonstrates the experimental results. Section 7 and 8 present the literature and conclusions.

2 PROBLEM FORMULATION

In this section, we first describe the adversary model, then define our privacy notion, and finally provide a general overview of our proposed approach.

2.1 Adversary Model

Denote a video as $\mathcal{V}$ which is captured by a video owner, e.g., a hospital or a company equipped with CCTV surveillance, and an agency which captures the video on the street. Video $\mathcal{V}$ (all the frames) includes a set of n sensitive objects $\mathbb{O} = \{O_1, O_2, \cdots, O_n\}$ (e.g., humans and vehicles). Assume that the video owner would like to share $\mathcal{V}$ to an external party for analysis (viz. the adversary). To ensure privacy, our proposed VERRO (randomly) generates a synthetic video $\mathcal{V}^*$ which is close to $\mathcal{V}$, such that:

- Each sensitive object in all the frames satisfies *ϵ-Object Indistinguishability* – the adversary cannot distinguish any two objects from the output synthetic video $\mathcal{V}^*$ with arbitrary background knowledge.
- The synthetic video $\mathcal{V}^*$ retains good utility (close to $\mathcal{V}$).

In VERRO, we assume that the adversaries can possess arbitrary background knowledge on each object (e.g., object contents, trajectories, at-scene times, and gathering groups of objects). To retain the output utility, VERRO does not change the background scene(s), but the privacy model can break the linkage between each object and the background scene(s) via *indistinguishability.*

With privacy guarantee for all the objects (making them indistinguishable), VERRO regularly generates synthetic videos for videos including sensitive objects w.r.t. multiple individuals (e.g., pedestrians and vehicles). In case that a video includes only one sensitive object, the adversary still cannot re-identify the object (see Section 5). In addition, VERRO only addresses the visual privacy concerns, assuming that the adversary cannot identify objects from the audio or audio is not captured (e.g., traffic monitoring and video surveillance).

2.2 Privacy Notion

2.2.1 Traditional Privacy Model. Video $\mathcal{V}$ includes multiple sensitive objects $O_1, \ldots, O_n$, which can be detected and tracked in all the frames [48, 49]. Specifically, it first detects all the sensitive objects in each frame with the detection algorithms (e.g., HOG for human [51] and SVM for vehicles [22]). Each detected object can be accurately tracked with the same ID if they highly overlap in multiple frames.

The traditional privacy models are defined to blur all the detected objects [1, 6, 20, 24, 42]. An alternative solution could be replacing the detected objects with "synthetic objects" [28, 43]. Each object can be replaced by a unique synthetic object: for instance, a red synthetic human and a purple synthetic human can be used to represent two different pedestrians in all the frames involving them. Then, the inferences and re-identification visually from the objects can be greatly mitigated.

2.2.2 ϵ-Object Indistinguishability for Sensitive Objects. Recall that only replacing the objects with synthetic objects in the video cannot address the re-identification based on the adversaries' background knowledge (as discussed in Section 1). Thus, we need to ensure *indistinguishability* for not only objects themselves (can be achieved by synthetic objects) but also their moving trajectories [50] in the video.

To this end, inspired from the indistinguishability provided by the ϵ-LDP, we define a novel privacy notion *ϵ-Object Indistinguishability* by considering each object's trajectory in the video (coordinates at different frames) as its "local data". Specifically, in the standard LDP definition [4, 10, 16], there are a set of users, each of which has its own data. After each user locally perturbs its data, the obfuscated output can be directly disclosed to any untrusted recipient/aggregator, where the randomized data collected from any two different users are indistinguishable [10, 16]. Migrating the LDP model to the objects in any video $\mathcal{V}$, we define the *ϵ-Object Indistinguishability* as below:

Definition 2.1 (ϵ-Object Indistinguishability). A randomization algorithm $\mathcal{A}$ satisfies *ϵ-Object Indistinguishability*, if and only if

[1] As formally defined in Definition 2.1, *ϵ-Object Indistinguishability* ensures a similar privacy guarantee as *ϵ-local differential privacy* [4, 10, 16].

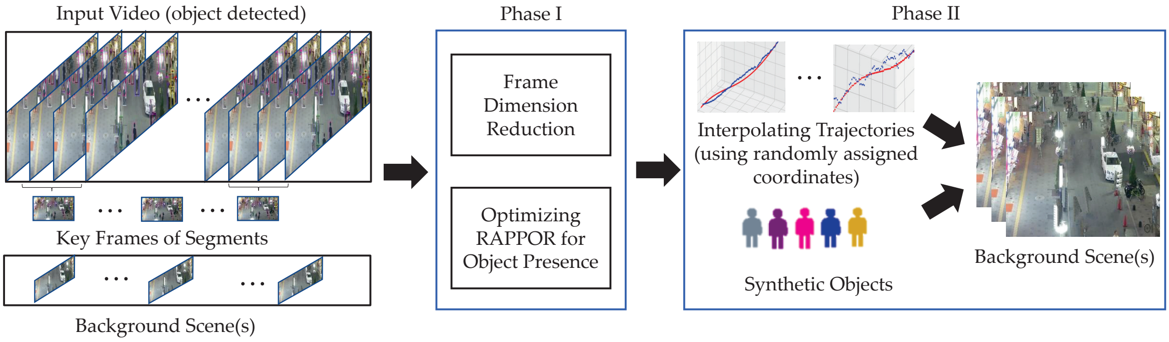

Input Video (object detected)
Key Frames of Segments
Background Scene(s)
Phase I
Frame Dimension Reduction
Optimizing RAPPOR for Object Presence
Phase II
Interpolating Trajectories (using randomly assigned coordinates)
Synthetic Objects
Background Scene(s)

Figure 2: VERRO for Utility-Driven Synthetic Video Generation with Object Indistinguishability

for any two input objects $O_i, O_j \in \mathbb{O}$ in the input video $\mathcal{V}$, and for any output object of $\mathcal{A}$ in the synthetic video $\mathcal{V}^*$ (denoted as y), we have $Pr[\mathcal{A}(O_i) = y] \leq e^\epsilon \cdot Pr[\mathcal{A}(O_j) = y]$.

Similar to ϵ-LDP [16], ϵ-*Object Indistinguishability* also focuses on the indistinguishability of randomizing any two objects, rather than the indistinguishability of randomizing any two neighboring inputs (whether any object is included or not included in the input) in traditional differential privacy [13]. Privacy budget ϵ decides the degree of indistinguishability (identical to LDP [16]).

Definition 2.1 guarantees that the randomly perturbed output of any two objects in $\mathcal{V}$ (both the object contents and the trajectories in all the frames) are ϵ-indistinguishable in $\mathcal{V}^*$. It also ensures *plausible deniability* for every object [5]. Since ϵ-*Object Indistinguishability* also requires all the objects to be visually indistinguishable (object contents), VERRO randomly assigns synthetic objects (e.g., the same shape but different colors) to replace the original distinct objects while generating the synthetic video $\mathcal{V}^*$. The synthetic objects are generated and placed by considering the distance of the object to the camera (e.g., the synthetic object size is larger if getting closer to the camera) [31].

2.3 VERRO Framework

The major components of VERRO (see Figure 2) consist of:

(1) **Preprocesssing**: all the objects are detected and tracked, and background scene (for each frame) is extracted using computer vision techniques [11, 48, 49].

(2) **Phase I**: for each object, its presence or absence in different frames/segments of the video are randomly generated (by random response) to be indistinguishable. Before executing random response, VERRO reduces the frame dimension in the video by detecting the key frames. Then, the utility can be improved by allocating optimal budgets for different dimensions. Furthermore, we also formulate a utility maximizing random response problem (optimizing RAPPOR [16]) to retain the optimal object presence information after Phase I. Note that Phase I satisfies ϵ-*Object Indistinguishability*: all the objects' presence in all the frames are indistinguishable (see details in Section 3).

(3) **Phase II**: with the randomly generated presence/absence information for each object, VERRO generates the synthetic video by inserting the synthetic objects into the video (background scene(s)). Specifically, the coordinates (where to insert the synthetic objects) are assigned, and computer vision techniques are applied to interpolate object moving trajectories between two assigned coordinates in the synthetic video. We also shown that Phase II does not leak any additional information (as a post-processing step [14]), and then VERRO satisfies ϵ-*Object Indistinguishability* (see details in Section 4).

3 PHASE I: OPTIMAL OBJECT PRESENCE

As the "local data" of each object (e.g., a pedestrian or vehicle) in the video $\mathcal{V}$, the object trajectory includes its presence or absence information in each frame and the coordinates in the frame (if present). In this section, we illustrate the Phase I of VERRO that first generates indistinguishable object presence.

3.1 Poor Utility with Random Response

We first define a bit vector for each object to indicate if such object is included in different frames or not:

Definition 3.1 (Object Presence Vector). Given video $\mathcal{V}$ which includes m different frames $F_1, \dots, F_m$ and n distinct objects $\mathbb{O} = \{O_1, ..., O_n\}$, whether each object $O_i, i \in [1, n]$ is present in frame $F_k, k \in [1, m]$ or not (all m frames) can form a bit vector: $B_i = (b_i^1, .., b_i^m) \in \{0, 1\}^m$ for object O_i.

It has been proven that a classic randomized response (RR) technique (e.g., RAPPOR [4, 16]) can be adapted to ensure ϵ-LDP for locally randomizing bit vectors. Similarly, a naive solution of ensuring ϵ-*Object Indistinguishability* for the object presence vectors is to directly the random response mechanism (*we will discuss how to optimize the utility in Section 3.2 and 3.3*). For each object $O_i \in \mathbb{O}, i \in [1, n]$, if object O_i exists in frame F_k, we set $b_i^k = 1, k \in [1, m]$. Otherwise, $b_i^k = 0$ holds in the vector B_i. Then, we flip one bit in vector $B_i, i \in [1, n]$ with a certain probability to report the true value. Then, all the perturbed bits in the object presence vector B_i can be combined as the output object presence vector for object O_i. Thus, the vectors $B_1, \dots, B_n$ (of all the objects) can be indistinguishable. Algorithm 1 shows the details of directly applying random response for object presence.

Algorithm 1 Random Response for Object Presence [16]

```
detect all the objects $\mathbb{O} = \{O_1, \dots, O_n\}$ in $\mathcal{V}$
for each $O_i, i \in [1, n]$ do
    collect the object presence vector $B_i = (b_i^1, .., b_i^m)$ in $\mathcal{V}$
    for each frame $F_k, k \in [1, m]$ do
        equally allocate budget $\epsilon/m$ to frame $F_k$
        random response for bit $b_i^k$ with the probability $\frac{e^{\epsilon/m}}{1+e^{\epsilon/m}}$
    end for
    $B_i \leftarrow (b_i^1, \dots, b_i^m)$
end for
Return $\forall i \in [1, n]$, $B_i$
```

Theorem 3.2. *Algorithm 1 randomly generates object presence vectors for objects with ϵ-Object Indistinguishability.*

Proof. *ϵ-Object Indistinguishability* can be proven by following the proof of ϵ-LDP with random response [16]. Given the object presence vectors $B_i = \{b_i^1, \ldots, b_i^m\}$ and $B_j = \{b_j^1, \ldots, b_j^m\}$ of any two objects $O_i, O_j \in \mathbb{O}$, for any possible output m-bit vector $y = (y^1, \ldots, y^m)$, we have:

$$\frac{Pr[\mathcal{A}(B_i) = y]}{Pr[\mathcal{A}(B_j) = y]} = \frac{Pr(b_i^1 = y^1)}{Pr(b_j^1 = y^1)} \cdots \frac{Pr(b_i^m = y^m)}{Pr(b_j^m = y^m)} \tag{1}$$

Since each bit is allocated with an equal privacy budget ϵ/m, the flipping probability would be $\frac{e^{\epsilon/m}}{1+e^{\epsilon/m}}$ [16]. For $k \in [1, m]$, if $b_i^k = b_j^k$ (either 0 or 1), then $\frac{Pr(b_i^k = y^k)}{Pr(b_j^k = y^k)}$ always equals 1. If $b_i^k \neq b_j^k$ and $b_i^k = y_k$, thus we have:

$$\frac{Pr(b_i^k = y^k)}{Pr(b_j^k = y^k)} = \frac{e^{\frac{\epsilon}{m}}}{1 + e^{\frac{\epsilon}{m}}} \cdot (1 + e^{\frac{\epsilon}{m}}) = e^{\frac{\epsilon}{m}} \tag{2}$$

Similarly, if $b_i^k \neq b_j^k$ and $b_j^k = y_k$, we have $\frac{Pr(b_i^k = y^k)}{Pr(b_j^k = y^k)} = e^{-\epsilon/m}$. Then, we have $\forall k \in [1, m]$, $\frac{Pr(b_i^k = y^k)}{Pr(b_j^k = y^k)} \leq e^{\epsilon/m}$ (equals one of 1, $e^{\epsilon/m}$ and $e^{-\epsilon/m}$). Combining all m bits, we have:

$$\frac{Pr[\mathcal{A}(B_i) = y]}{Pr[\mathcal{A}(B_j) = y]} \leq e^{\epsilon} \tag{3}$$

Thus, the generated presence bit vectors satisfy *ϵ-Object Indistinguishability*. This completes the proof. □

Poor Utility. Although Algorithm 1 satisfies *ϵ-Object Indistinguishability*, the utility of synthetic video would be extremely low since the total number of frames in a video m can be thousands or more, and then the allocated budget for each frame would be negligible. It destroys the utility of random response (i.e., RAPPOR [16]). For instance, a vehicle occurs in 100 frames out of a 1000-frame video, then the privacy budget for each frame is $\epsilon/1000$, which makes the flipping probability close to 0.5. Then, each of the 1000 frames would have 50% probability to include the vehicle (and other vehicles), then the objects in the video are too random (extremely low utility at this time). Thus, we explore an alternative solution for the video data in Section 3.2 and 3.3.

3.2 Dimension Reduction in the Video

Recall that the limited utility in Algorithm 1 results from the high dimensions in the video (considering each frame as a dimension). Most existing LDP techniques (e.g., RAPPOR [16], LDPMiner [40], and PLDP [9]) have reduced the dimension (e.g., bloom filter reduces the bits dimension for RAPPOR [16], top k frequent items reduces the dimension of items in LDPMiner [40], and Johnson-Lindenstrauss transform reduces the dimension of location data [9]). In videos, since difference between two consecutive frames is very small, we extract the key frames [12, 19, 30] out of m frames from $\mathcal{V}$ to reduce dimension in VERRO.

3.2.1 Key Frame Extration.

3.2.1 Key Frame Extration. In computer vision, many existing key frame extraction algorithms have been proposed based on the boundary method [19], motion analysis [12], clustering [30], among others. Since algorithms based on clustering has been shown to generate more accurate results [30], we integrate it into VERRO for dimension reduction. The basic idea is to divide the video into several groups of similar frames.

The algorithm [39] first transforms each pixel RGB value to construct the HSV (hue, saturation, value) histogram for each frame, and then calculates the pixel distribution in terms of hue, saturation, value, respectively. Each cluster is initialized with a new frame, and expanded by adding new consecutive frames which are similar to the existing frames (measured by the HSV histograms). After the clustering, each cluster includes a group of consecutive frames, which can be considered as a segment of the video. Finally, a key frame can be extracted from each cluster/segment. The details are illustrated in Algorithm 2.

As a result, the key frame can be utilized to represent every segment. Then, the m-bit object presence vectors (for all the objects) can be reduced to ℓ-bit vectors. For instance, key frames $\mathcal{F}_1, \ldots, \mathcal{F}_\ell$ (where ℓ denotes the number of key frames, and $\ell \ll m$ in general) are extracted from $\mathcal{V}$. Object O_i's presence vector B_i can be reduced to $B'_i = (kb_i^1, \cdots, kb_i^\ell)$.

Algorithm 2 Segmentation and Key Frame Extraction

```
initialize the first segment $S_1 = F_1$, segment index $i = 1$
equally partition $H$, $S$, $V$ value ranges to $h$, $s$ and $v$ parts
for each frame $F_k$, $k \in [2, m]$ do
    for each part $\hat{h}$, $\hat{s}$, $\hat{v}$ in H, S, V do
        construct the histograms $H(\hat{h})$, $S(\hat{s})$, $V(\hat{v})$ in frame $F_k$
    end for
    $Sim_H(F_k, S_i) = \sum_{\hat{h}=1}^{h} \min\{H(\hat{h}), S_i[H(\hat{h})]\}$
    $Sim_S(F_k, S_i) = \sum_{\hat{s}=1}^{s} \min\{S(\hat{s}), S_i[S(\hat{s})]\}$
    $Sim_V(F_k, S_i) = \sum_{\hat{v}=1}^{v} \min\{V(\hat{v}), S_i[V(\hat{v})]\}$
    $\{\alpha, \beta, \gamma$: weights for H, S, V; similarity threshold: $\tau\}$
    if $(\alpha \cdot SI_H + \beta \cdot SI_V + \gamma \cdot SI_S) \geq \tau$ then
        $S_i \leftarrow S_i \cup F_k$
    else
        $i = i + 1$ and initialize a new segment $S_i$
        $S_i \leftarrow S_i \cup F_k$
    end if
end for
for each segment $S_i$ do
    compute the maximum frame entropy $Entropy(F)$:
    max $\{-\alpha \cdot \sum_{\hat{h}=1}^{h}[H(\hat{h}) \log H(\hat{h})] - \beta \cdot \sum_{\hat{s}=1}^{s}[S(\hat{s}) \log S(\hat{s})] - \gamma \cdot \sum_{\hat{v}=1}^{v}[V(\hat{v}) \log V(\hat{v})]\}$
    extract the key frame with maximum entropy $\mathcal{F}_i$ in $S_i$
end for
return all the segments and key frames
```

3.2.2 Random Response. After dimension reduction, random response can be implemented based on the RAPPOR framework [16] for each object. Each bit kb_i^k in ℓ-bit vector of object O_i is randomly flipped into 0 or 1 using the following rules:

$$kb_i^k = \begin{cases} kb_i^k, \text{ with the probability of } (1-f) \\ 1, \text{ with the probability of } \frac{f}{2} \\ 0, \text{ with the probability of } \frac{f}{2} \end{cases} \tag{4}$$

Theorem 3.3. *The random response (with rules in Equation 4) $\ell \log(\frac{2-f}{f})$-Object Indistinguishability.*

Proof. Again, object indistinguishability can be proven by following the proof of LDP [16]. Specifically, the RAPPOR [16] satisfies $2h \log(\frac{2-f}{f})$-LDP with the output size of the hash function in the bloom filter h and the flipping probability f. Maximum difference sizes are $2h$ between two input values. Thus, the random response (with rules in Equation 4) make ϵ equal to $\ell \log(\frac{2-f}{f})$ since size difference in any two presence vector is at most ℓ (by replacing the encoded bit vectors of bloom filter as the object presence vectors in RAPPOR [16]), which satisfies $\ell \log(\frac{2-f}{f})$-*Object Indistinguishability.* □

3.3 Optimizing RAPPOR for Object Presence

Although ℓ is far less than m, the number of key frames ℓ may still be large depending on the background scene(s), activity motion and light density. To solve this, we can further reduce the dimension by choosing a subset of key frames out of ℓ key frames to allocate the privacy budget. Indeed, determining whether each key frame is picked for allocating the privacy budget or not can be formulated as an optimization problem (*maximizing the utility of generating the synthetic video using the random object presence vectors in Phase II*).

3.3.1 Optimization Problem. For each key frame $\mathcal{F}_k, k \in [1, \ell]$, we define a binary variable $x_k \in \{0, 1\}$ to represent if key frame $\mathcal{F}_k$ is picked for budget allocation or not. Then, the total number of picked key frames is referred as $\sum_{k=1}^{\ell} x_k$. Per the Theorem 3.3, we have the random response satisfies $\sum_{k=1}^{\ell} x_k \log(\frac{2-f}{f})$-*Object Indistinguishability.*

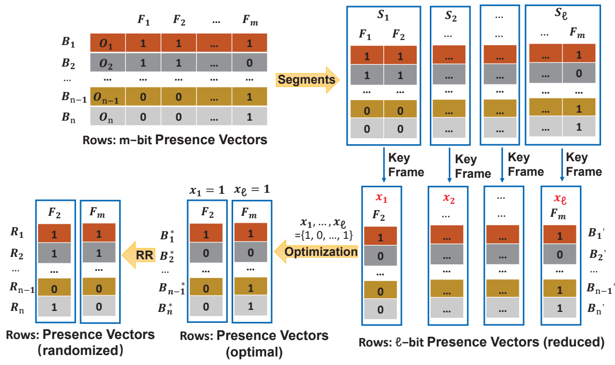


Figure 3: Dimension Reduction, Utility Maximization and Random Response

An example for dimension reduction, utility maximization and random response is given in Figure 3. Considering the n objects $O_1, \dots, O_n$, after dimension reduction, all the n object presence vectors are reduced to n different ℓ-bit vectors. Our goal is to accurately retain more objects in the video, thus we aim at minimizing the distance between $\forall i \in [1, n], B_i'$ (extracted from $\mathcal{V}$) and $\forall i \in [1, n], R_i$ (denoted as the ℓ-bit vectors by applying random response to $\forall i \in [1, n], B_i'$).

Specifically, since $\forall i \in [1, n], R_i$ are randomized bit vectors (the kth entry in all the vectors are 0 if $x_k = 0$), we should measure the difference between the expectation $\forall i \in [1, n], E(R_i) = E[(R_i^1, \dots, R_i^{\ell})]$ and $B_i' = (kb_i^1, \dots, kb_i^{\ell})$.

We first learn the expectation of R_i^k (the kth entry in R_i). If $x_k = 0$, then $\forall i \in [1, n], R_i^k = 0$ hold. Thus, we have:

$$E(R_i^k) = x_k \cdot [Pr(R_i^k = 1) \cdot 1 + Pr(R_i^k = 0) \cdot 0] \tag{5}$$

There are two cases for R_i^k (in case of $x_k = 1$):

(1) If $kb_i^k = 1$, per Equation 4, we have $E[R_i^k] = 1 \cdot [(1 - f) \cdot 1 + \frac{f}{2} \cdot 0 + \frac{f}{2} \cdot 1]$.

(2) If $kb_i^k = 0$, we have $E[R_i^k] = 1 \cdot [(1 - f) \cdot 0 + \frac{f}{2} \cdot 0 + \frac{f}{2} \cdot 1]$.

Thus, the expectation can be summarized as following:

$$\begin{cases} E(R_i^k) = \frac{f}{2}, \text{ if } x_k = 1 \text{ and } kb_i^k = 0 \\ E(R_i^k) = 1 - \frac{f}{2}, \text{ if } x_k = 1 \text{ and } kb_i^k = 1 \\ E(R_i^k) = 0, \text{ if } x_k = 0 \text{ and } kb_i^k = 0 \text{ or } 1 \end{cases} \tag{6}$$

The objective function can be formulated as:

$$\min : \sum_{k=1}^{\ell} [x_k \sum_{i=1}^{n} |E(R_i^k) - kb_i^k|] \tag{7}$$

Furthermore, for accurately interpolating the objects in different frames in Phase II, the number of key frames picked for budget allocation should be no less than 2. Therefore, we formulate the optimization problem as below:

$$\begin{aligned} &\min : \sum_{k=1}^{\ell} [x_k \sum_{i=1}^{n} |E(R_i^k) - kb_i^k|] \\ &s.t. \begin{cases} 2 \leq \sum_{k=1}^{\ell} x_k \leq \ell \\ \forall k \in [1, \ell], x_k \in \{0, 1\} \end{cases} \end{aligned} \tag{8}$$

Detailing expectation $E(R_i^k)$ with the flipping probability, the optimization problem can be converted to:

$$\begin{aligned} &\min : \sum_{k=1}^{\ell} (x_k |\frac{n \cdot f}{2} - f \cdot \sum_{i=1}^{n} kb_i^k|) \\ &s.t. \begin{cases} 2 \leq \sum_{k=1}^{\ell} x_k \leq \ell \\ \forall k \in [1, \ell], x_k \in \{0, 1\} \end{cases} \end{aligned} \tag{9}$$

3.3.2 Complexity and Solver. Since f and $\forall k \in [1, \ell], \forall i \in [1, n], kb_i^k$ are constants, $\forall k \in [1, \ell], |\frac{n \cdot f}{2} - f \cdot \sum_{i=1}^{n} kb_i^k|$ are constants. Then, Equation 9 is a binary integer programming (BIP) problem. Although solving the BIP problems can be NP-hard [27], we can approximately solve Equation 9 using linear programming (LP) since the objective function and the constraints are *linear*: (1) letting the binary variable $\forall k \in [1, \ell], x_k$ be continuous in $[0, 1]$, (2) solving the problem using standard LP solvers (e.g., the Simplex algorithm), and (3) in the optimal solution of the LP problem, $\forall k \in [1, \ell]$, if $x_k \in [0, 0.5)$, we assign $x_k = 0$; if $x_k \in [0.5, 1]$ we assign $x_k = 1$ as the approximated optimal solution of the BIP problem.

3.3.3 Addressing Possible Privacy Leakage in Optimization. Compared to randomly picking a number of key frames for budget allocation, computing the optimal frames for budget allocation may result in some minor privacy leakage since the total number of objects in the kth key frame $\sum_{i=1}^{n} kb_i^k, k \in [1, \ell]$ (which is used in the optimization) might be different. Such privacy leakage is generally minor due to a small sensitivity Δ of the object count in each frame (e.g., $\Delta = 1$ for protecting the presence/absence of each object in every frame). Thus, it can be addressed by injecting a small amount of generic Laplace noise $Lap(\frac{\Delta}{\epsilon'})$ into $\sum_{i=1}^{n} kb_i^k, k \in [1, \ell]$ before formulating the optimization problem. Although adding such small amount of noise may slightly deviate the optimality, this could guarantee end-to-end indistinguishability (differential privacy). Since such privacy guarantee is well studied in literature [13], we do not discuss it in this paper due to space limitation.

3.4 Privacy Guarantee

After solving the optimization problem, as shown in Figure 3, each of the picked key frames will be allocated with a privacy budget $\epsilon / \sum_{k=1}^{\ell} x_k$. In the meanwhile, VERRO utilizes the optimal solution $\forall k \in [1, \ell], x_k$ n to derive the optimal presence vectors ($\sum_{k=1}^{\ell} x_k$-bit), denoted as $B_1^*, \dots, B_n^*$. Next, random response is applied to $B_1^*, \dots, B_n^*$ to generate output presence vectors $R_1, \dots, R_n$.

Theorem 3.4. *Phase I satisfies ϵ-Object Indistinguishability.*

Proof. Phase I derives the presence bit vectors B_i^* and B_j^* for any two objects O_i and O_j after the optimization. Then, random response is applied to B_i^* and B_j^* and generate random vectors R_i and R_j. Per Theorem 3.3, Phase I satisfies *ϵ-Object Indistinguishability* where $\epsilon = \sum_{k=1}^{\ell} x_k \ln \frac{2-f}{f}$ (note that the privacy guarantee for utility maximization has been discussed in Section 3.3.3). □

It is worth noting that the presence of objects in the remaining $(m - \sum_{k=1}^{\ell} x_k)$ frames and the coordinates of the objects in all m frames in the synthetic video $\mathcal{V}^*$ will be generated in Phase II.

4 PHASE II: VIDEO GENERATION

In this section, we illustrate the details of Phase II.

4.1 Background Scene(s)

As discussed in Section 2, video preprocessing includes detecting/tracking objects and background scene(s) extraction. While removing objects from digital images (e.g., each frame of a video), the pixels within the objects are missing in the frame and need to be reconstructed for the background scene(s). In VERRO, we utilize an efficient algorithm [11] to fill the blank area by considering both texture and structure.

First, the quality of the output image/frame highly depends on the order of filling different parts of the blank areas. The algorithm provides a filling strategy by prioritizing them using the combination of the continuation of strong edges and high-confidence surrounded pixels. The priority is computed for every border patch, with distinct patches for each pixel on the boundary of the blank areas. Then, we always start filling at the border pixels with the highest priority.

Second, while filling the pixel p, the algorithm places it at the centroid of a patch with certain size (e.g., 3×3). Then, we traverse all the background pixels, and the centroid pixel of the most similar patch from the source background region will be filled in p, where the similarity is measured by the sum of squared errors. Some reconstructed background scenes are demonstrated in Section 6.

4.2 Randomly Generating Object Coordinates

Phase I generates indistinguishable presence information (in different frames) for all the objects. Next, we need to insert synthetic objects into the background scene (each frame) to generate the synthetic video $\mathcal{V}^*$. Specifically, we denote all the frames in the synthetic video $\mathcal{V}^*$ as $\{F_1^*, \dots, F_m^*\}$, and the frames in $\mathcal{V}^*$ corresponding to the original key frames as $\{\mathcal{F}_1^*, \dots, \mathcal{F}_\ell^*\}$. We then discuss different cases of generating coordinates for the objects in each frame.

4.2.1 $R_i = \emptyset$. If all the entries in any object presence vector are 0, such random vector output R_i would result in object loss (the synthetic video will lose one object), and it is unnecessary to identify the coordinates for them in this case. We have evaluated such utility loss in Section 6, and most of the objects can be retained by VERRO in practice.

4.2.2 $R_i \neq \emptyset$. If there exists at least one non zero entry in R_i, then an object will be inserted to the synthetic video $\mathcal{V}^*$. A critical and challenging question is that where to insert the object. We employ the coordinates of all the objects in the original video $\mathcal{V}$ as "Candidate Coordinates" to generate the coordinates in each frame of the synthetic video.

Specifically, in each key frame of the synthetic video $\forall k \in [1, \ell], \mathcal{F}_k^*$, the number of objects inserted into key frame $\mathcal{F}_k^*$ is $\sum_{i=1}^{n} R_i^k$ (derived in Phase I). Denoting the number of objects in the kth key frame of $\mathcal{V}$ as $c_k, k \in [1, \ell]$ where $c_k = 0$ if $x_k = 0$, we thus have:

- *Sufficient candidate coordinates*: if $\sum_{i=1}^{n} R_i^k \leq c_k$, the number of required objects in $\mathcal{F}_k^*$ is no greater than the number of candidate coordinates in $\mathcal{F}_k$. Then, VERRO randomly picks $\sum_{i=1}^{n} R_i^k$ out of c_k candidate coordinates for $\sum_{i=1}^{n} R_i^k$ different objects in the background scene (frame $\mathcal{F}_k^*$). Please see the left example in Figure 4.
- *Insufficient candidate coordinates*: if $\sum_{i=1}^{n} R_i^k > c_k$, the number of required objects in $\mathcal{F}_k^*$ is greater than the number of candidate coordinates in $\mathcal{F}_k$. For instance, in the right example in Figure 4, we expand the set of candidate coordinates by adding the candidate coordinates in $\mathcal{F}_k$'s neighboring frames in the same segment. Then, VERRO randomly picks $\sum_{i=1}^{n} R_i^k$ out of c_k' candidate coordinates (c_k' is expanded from c_k where $c_k < \sum_{i=1}^{n} R_i^k \leq c_k'$) to insert $\sum_{i=1}^{n} R_i^k$ different objects into the background scene (frame $\mathcal{F}_k^*$).

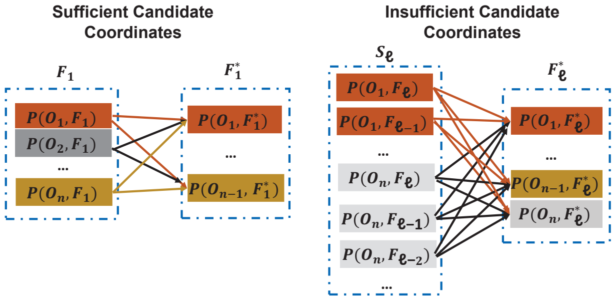


Figure 4: Random Coordinates Assignment (before Interpolation)

After assigning coordinates to the key frames (where $R_i^k = 1$), we obtain at least 1 frame with the corresponding coordinates for any O_i (if the corresponding object is retained in the synthetic video) – the retained object has been assigned with coordinates in at least two frames in almost all the cases in our experiments in Section 6. With such randomly assigned coordinates in some key frames, we can interpolate the coordinates in other frames (out of m frames in total) between such key frames. For instance, given coordinates in two key frames F_1 and F_{10} for object O_i, then its coordinates between F_1 and F_{10} can be estimated. In literature, there are many interpolation methods for moving object trajectories data, such as nearest neighbor interpolation [21] and Lagrange interpolation [17]. In VERRO, we adopt the Lagrange interpolation to estimate such trajectories with the randomly generated positions.

Finally, after interpolation, we define the first frame in which any object first occurs as "head" and the frame where such object last occurs as "end" in the interpolated trajectory. The head and end generally involve such object on the border of the frame. Thus, the interpolation terminates as each object's head and end are identified on the border of the frame (*objects do not occur in all the frames in general*).

Theorem 4.1. *VERRO (Phase I and Phase II) satisfies ϵ-Object Indistinguishability.*

Proof. Given any two objects O_i and O_j, their randomly generated presence vectors R_i and R_j are proven to be *ϵ-Object Indistinguishable* (after Phase I). We now examine the randomly assigned coordinates in the key frames and two full interpolated trajectories in the synthetic video $\mathcal{V}^*$.

Specifically, given any output presence vector y and any output trajectory $t = \{t_1, \ldots, t_m\}$ in $\mathcal{V}^*$, for simplicity of notation, we also denote the trajectories of O_i and O_j in $\mathcal{V}^*$ as $O_i = \{T_i^1, \ldots, T_i^m\}$ and $O_j = \{T_j^1, \ldots, T_j^m\}$, respectively.

$$\frac{Pr[\mathcal{A}(O_i) = t]}{Pr[\mathcal{A}(O_j) = t]}$$
$$= \frac{Pr[\mathcal{A}(B'_i) = y]}{Pr[\mathcal{A}(B'_j) = y]} \cdot \frac{Pr[\mathcal{A}(T_i^1) = t_1]}{Pr[\mathcal{A}(T_j^1) = t_1]} \cdots \frac{Pr[\mathcal{A}(T_i^m) = t_m]}{Pr[\mathcal{A}(T_j^m) = t_m]}$$

On one hand, we have $\frac{Pr[\mathcal{A}(B'_i)=y]}{Pr[\mathcal{A}(B'_i)=y]]} \leq e^{\epsilon}$ (Phase I). On the other hand, if $\forall k \in [1, m], R_i^k = R_j^k = 1$, two objects are present in the same frame F_k (and F_k^*). In this case, since the same randomization is applied to O_i and O_j to pick the coordinates from the same set of candidates, we have $\forall k \in [1, m], Pr[\mathcal{A}(T_i^k) = t_k] = Pr[\mathcal{A}(T_j^k) = t_k]$. If $\forall k \in [1, m], R_i^k = R_j^k = 0$ (the coordinates are interpolated from the coordinates randomly assigned in the previous case [14]), we also have $\forall k \in [1, m], Pr[\mathcal{A}(T_i^k) = t_k] = Pr[\mathcal{A}(T_j^k) = t_k]$.

To sum up the above three cases, we have:

$$\frac{Pr[\mathcal{A}(O_i) = t]}{Pr[\mathcal{A}(O_j) = t]} \leq e^{\epsilon} \tag{10}$$

where $\epsilon = \sum_{k=1}^{\ell} x_k \log(\frac{2-f}{f})$, as analyzed in Theorem 3.3 and Section 3.3. This completes the proof. □

Finally, we summarize the procedures and privacy guarantee in VERRO. Given an video, the presence of objects in all the frames are indistinguishable via random response. Then, adversaries cannot identify specific objects by the frame presences with any background knowledge. Furthermore, we randomly generate synthetic positions of objects. Therefore, we claim that any object in the input $\mathcal{V}$ can possibly generate any object in the output $\mathcal{V}^*$ (with random response in Phase I and random coordinates assignment in Phase II).

5 DISCUSSION

Distributed Framework: LDP techniques [4, 16] are deployed in distributed setting where each user perturbs its local data to share. Our object-based privacy model ensures indistinguishability at the object level where all the "distributed" local data can be perturbed by a "local agent" (aka. video owner) and shared as $\mathcal{V}^*$ to untrusted recipients.

Different video owners can also share their perturbed videos to any untrusted recipient (all the objects in each video are still well protected). Note that VERRO does not ensure video level indistinguishability (all the videos are indistinguishable). We will investigate the utility of the video level indistinguishability in practice and explore the LDP solutions in the future.

Noise Cancellation: in VERRO, objects and their trajectories are generated in the sanitized video. Thus, the individual noises resulted from random response for all the objects may not be directly canceled in the output video. Indeed, after random response and random coordinates assignment, there exists trajectories in the sanitized video which are close to the original trajectories (as shown in Figure 6-8 in our experiments). Also, such noise can be cancelled in data aggregation applications [9] (e.g., object counting, as shown in Figure 12 and 13).

Multiple Object Types: in this paper, we use pedestrians and vehicles as concrete examples to show their indistinguishability in the publishable synthetic video. It is worth noting that other objects can also be protected with the defined privacy notion in VERRO by replacing the detecting algorithms and synthetic objects. Furthermore, if any video includes multiple types of objects (e.g., pedestrians and vehicles), VERRO can generate the synthetic video for different types of objects, respectively. For instance, it first randomly generates pedestrians, and then randomly generates the vehicles. All the pedestrians are *ϵ-Object Indistinguishable* while all the vehicles are *ϵ-Object Indistinguishable*, assuming that it does not leak additional information across different object types (as all the objects have been replaced with random synthetic objects in the same type).

Protection for One-Object Video: VERRO can generate synthetic videos in which all the objects are ϵ-indistinguishable. In case that the video includes only one sensitive object, VERRO can also protect such object against re-identification. In existing LDP techniques [4, 16], if only one user perturbs its Object data and discloses it to the untrusted aggregator, the original data cannot be identified from its perturbed data. Similar to such works (e.g., RAPPOR [16]), the objects and the trajectories cannot be identified from the perturbed presence in the synthetic video even if the adversary has arbitrary background knowledge on the presence of individuals at specific times.

Imperfect Background Scene(s): as discussed in Section 4, background scene(s) is extracted from the original video. The reconstructed scene may not be as perfect as the original frame (e.g., human/vehicle silhouette or duplicated/blurred region may occur). Thus, imperfect background scene(s) may leak some privacy about "there exists some object in the silhouette or blurred regions in the original video". However, adversaries cannot infer that "who is in that region or which object is in that region" since all the objects are indistinguishable from end to end.

System Deployment: the proposed VERRO can be implemented as an application, and deployed as a component to generate utility-driven synthetic videos by processing the videos captured by each camera (e.g., in the surveillance system, integrated with the traffic monitoring facilities, in smart phones or other mobile devices) where *ϵ-Object Indistinguishability* can be guaranteed.

6 EXPERIMENTS

In this section, we present the performance evaluations.

6.1 Experimental Setup

We conduct our experiments on three real videos in the repository of multiple object tracking benchmark[2]. To benchmark the results, we choose three pedestrian videos, two videos are captured by static cameras while the third video is recorded by a moving camera (where multiple background scenes are extracted):

(1) MOT16-01 (people walking around a large square, denoted as "MOT01") [35]: 23 distinct pedestrians are sensitive objects in 450 frames (static camera).
(2) MOT16-03 (pedestrians on the street at night, denoted as "MOT03") [35]: 148 distinct pedestrians are sensitive objects in 1,500 frames (static camera).
(3) MOT16-06 (street scene from a moving platform, denoted as "MOT06") [35]: 221 distinct pedestrians are sensitive objects in 1,194 frames (moving camera).

Table 1: Characteristics of Experimental Videos

Video	Resolution	Frame #	Objects	Camera
MOT16-01	1920 × 1080	450	23	static
MOT16-03	1920 × 1080	1,500	148	static
MOT16-06	640 × 480	1,194	221	moving

We implement the detecting/tracking algorithm [48, 49] to identify all the objects (pedestrians). Objects are detected in each frame, and the same object is marked with the same ID in the entire video. Computer vision technique [11] is also utilized to extract/reconstruct the background scene(s) from the input video $\mathcal{V}$. All the programs are implemented in Python 3.6.4 with the OpenCV 3.4.0 library and tested on an HP PC with Intel Core i7-7700 CPU 3.60GHz and 32G RAM.

6.2 Generic Utility Evaluation

We first evaluate the utility of our synthetic videos. The proposed VERRO is a two-phase LDP approach. In Phase I, it randomly generates the object presence in all the frames of the synthetic video ("1" or "0"). In Phase II, we interpolate the trajectories. Thus, we evaluate two different types of utility: (1) the retained utility after Phase I (Random Response), and (2) the utility of synthetic video after Phase II.

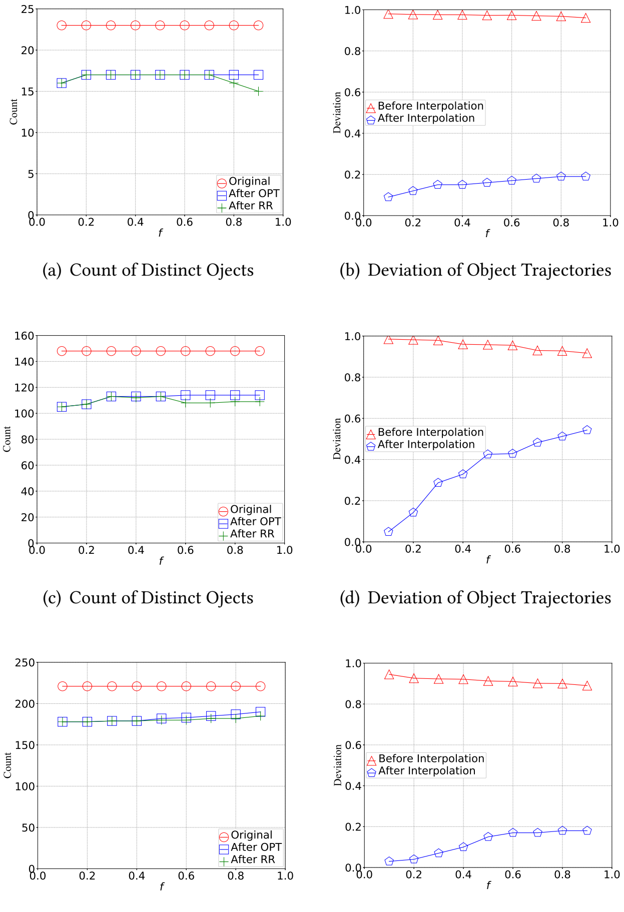
6.2.1 Utility for Phase I. Phase I generates "presence bit vectors" for all the objects with frame dimension reduction, optimization ("OPT") and random response ("RR"). Some objects might not be included in the key frames, and/or might not be generated in the random response. Then, such objects cannot be generated in the synthetic video (all the entries in the corresponding vectors are 0) since they cannot be interpolated without any object presence in Phase I (also treated as noise). Thus, we evaluate the count of distinct objects (pedestrians) in Phase I.

First, Table 2 shows some results after detecting key frames for frame dimension reduction. In video MOT01, there are 22 key frames, and 19 out of 23 objects are present in the key frames. In video MOT03, 52 key frames are extracted, and 124 out of 148 objects are present in such key frames. In video MOT06, 191 out of 221 objects are captured in the identified 48 key frames. We can observe that frame dimension reduction results in less utility loss (retaining $\sim 80\%$ distinct objects).

Figure 5(a), 5(c) and 5(e) present the count of distinct objects in original video, after optimization ("OPT"), and random response ("RR"). We set the flipping probability f from 0.1 to 0.9 for random response. In Figure 5(a), approximately 17 distinct objects can be retained in 10 key frames (optimized). f only slightly affects the optimization: the count of distinct objects increases a little bit as f grows. To evaluate how f affects the random response, we can observe that one or two objects are not randomly generated in RR as f grows to a large flipping probability (e.g., 0.8). This matches the fact that higher f results in worse utility in random response (Theorem 3.2) – such utility loss is indeed minor in our experiments. In addition, we can draw similar observations in Figure 5(c) and 5(e) where the utility loss of random response is even less for videos MOT03 and MOT06. Thus, Phase I retains a high percent of distinct objects via their random presence vectors, which means less side effect introduced by RR (this facilitates the interpolation in Phase II for boosting utility).

Table 2: Distinct Objects after Key Frame Extraction

Video	Frame #	Objects #	Key Frame #	Remaining #
MOT01	450	23	22	19
MOT03	1,500	148	52	124
MOT06	1,194	221	48	191



(a) Count of Distinct Ojects

(b) Deviation of Object Trajectories

(c) Count of Distinct Ojects

(d) Deviation of Object Trajectories

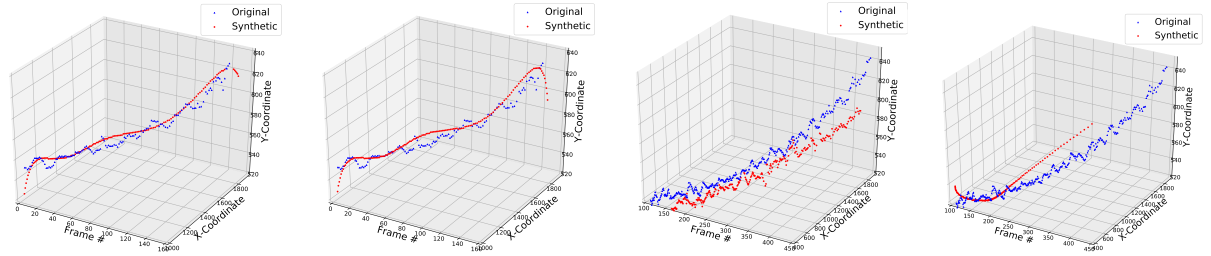
(e) Count of Distinct Ojects

(f) Deviation of Object Trajectories

Figure 5: Utility Evaluation of Phase I & II of MOT01 (MOT03 and MOT06)

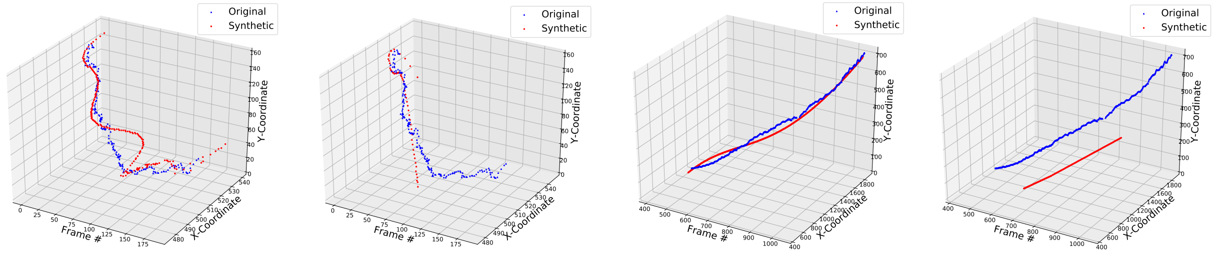
6.2.2 Utility for Phase II. Since the synthetic video generated in Phase II includes the synthetic objects at the same scene, the corresponding synthetic object of each original object (e.g., pedestrian) may have different coordinates in the same frame.

[2] https://motchallenge.net/



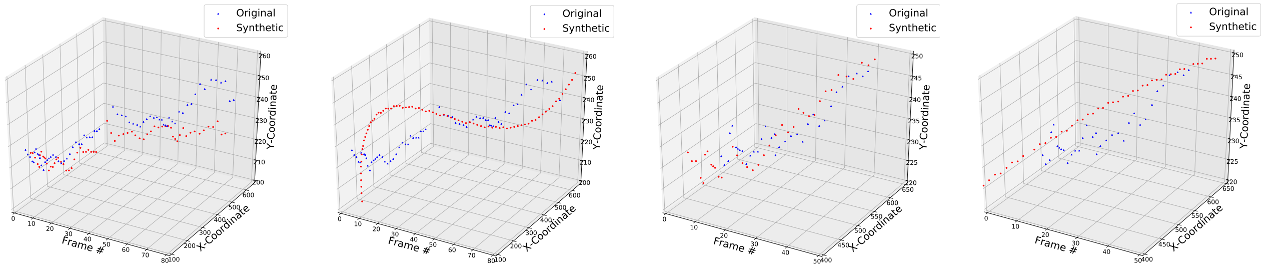
(a) Object #2 (f=0.1) (b) Object #2 (f=0.9) (c) Object #9 (f=0.1) (d) Object #9 (f=0.9)

Figure 6: Trajectories of Two Randomly Selected Objects in MOT01



(a) Object #35 (f=0.1) (b) Object #35 (f=0.9) (c) Object #105 (f=0.1) (d) Object #105 (f=0.9)

Figure 7: Trajectories of Two Randomly Selected Objects in MOT03



(a) Object #5 (f=0.1) (b) Object #5 (f=0.9) (c) Object #165 (f=0.1) (d) Object #165 (f=0.9)

Figure 8: Trajectories of Two Randomly Selected Objects in MOT06

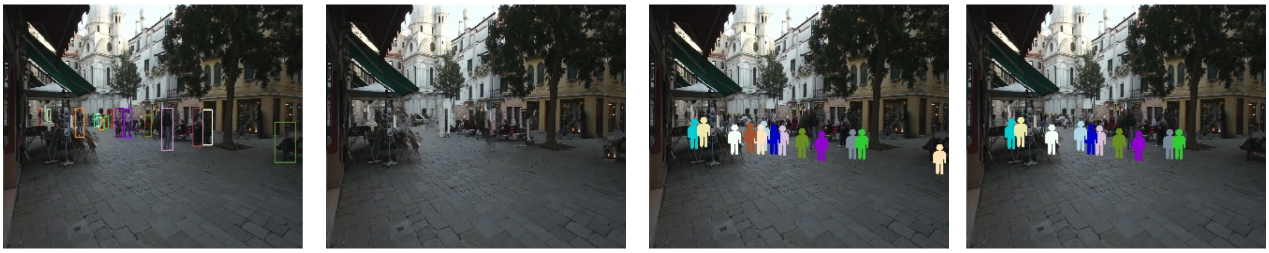
All the coordinates in different frames may form a trajectory in the synthetic video. Thus, we also measure the deviation for the trajectories of all the objects in the original video and synthetic video: $\sum_{i=1}^{n}\sum_{k=1}^{m}\frac{P(O_i,F_k)-P(O_i,F_k^*)}{P(O_i,F_k)}$, where $P(O_i, F_k)$ and $P(O_i, F_k^*)$ are the center coordinates of object O_i in the kth frame of the input video and the synthetic video.

In Figure 5(b), 5(d) and 5(f), we can observe that the deviation before Phase II is higher than 0.9, since each object is only generated in a few frames. The deviation of trajectories increases as the flipping probability f gets larger since more flips occur more frequently (e.g., "0" to "1", or vice-versa). In such three figures, after Phase II, the deviation can be significantly reduced (e.g., in $[0.1, 0.2]$ for video MOT01, in $[0.02, 0.2]$ for video MOT06).

More specifically, we randomly select two objects (e.g., pedestrians) from each of the three videos, and extract their trajectories in the original video $\mathcal{V}$. In addition, we also extract their corresponding trajectories in the synthetic video $\mathcal{V}^*$. Figure 6, 7 and 8 demonstrate the trajectories of those objects in the input videos and synthetic videos, where 3-dimensional axes refer to the frame ID and coordinates (X, Y) in videos. As $f = 0.1$, the trajectories of the objects lie closer to the original ones (compared to $f = 0.9$). It is worth noting that any object (pedestrian) in the original video can generate the corresponding trajectory of any object (e.g., the plotted trajectories corresponding to Object #2 and Object #9 in Figure 6). This is ensured by the ϵ-indistinguishable presence bit vectors randomly generated from all the objects in VERRO.

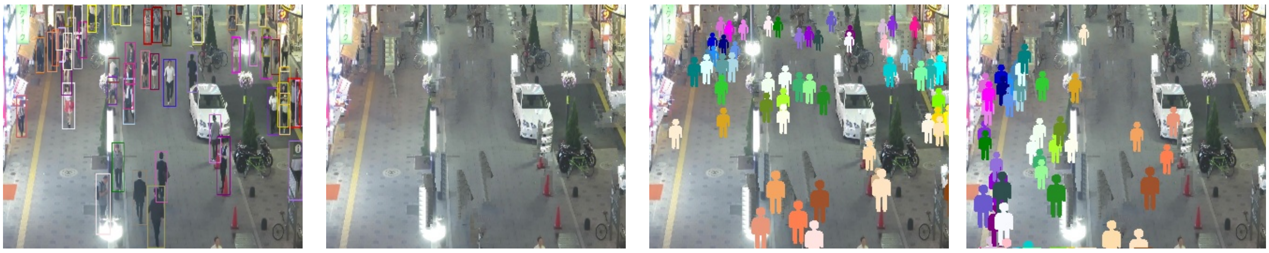
6.3 Visual & Aggregated Results

We also randomly pick a frame from each of the three experimental videos, and present the generated background scenes and the corresponding frames in the synthetic videos. For video MOT01, Figure 9(a) shows the input frame and the detected objects in the frame. Also, we use a background interpolation algorithm [11] to fill the missing pixels (after removing all the detected objection), as shown in Figure 9(b). Similarly, a randomly picked frame (with the detected objects) and the generated background



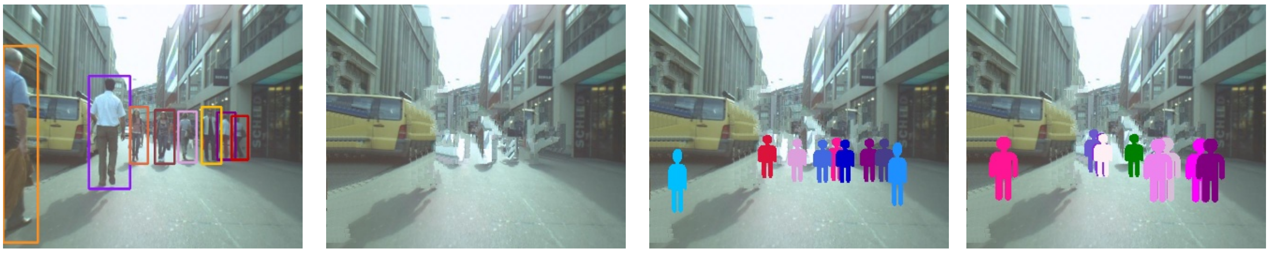
(a) Frame 8 (b) Background Scene (c) Synthetic Frame (f=0.1) (d) Synthetic Frame (f=0.9)

Figure 9: Representative Frames in MOT01 and the Generated Synthetic Video



(a) Frame 134 (b) Background Scene (c) Synthetic Frame (f=0.1) (d) Synthetic Frame (f=0.9)

Figure 10: Representative Frames in MOT03 and the Generated Synthetic Video



(a) Frame 216 (b) Background Scene (c) Synthetic Frame (f=0.1) (d) Synthetic Frame (f=0.9)

Figure 11: Representative Frames in MOT06 and the Generated Synthetic Video

scenes in MOT03 and MOT06 are given in the first two subfigures of Figure 10 and 11. Some human silhouettes still exist in the background scenes. Clearly, the silhouettes cannot be associated to any objects in the synthetic video (as shown in Figure 10(c), 10(d), 11(c) and 11(d)). This confirms the discussion for imperfect background scene in Section 5.

In the synthetic videos, we use different colors for different synthetic objects. Compare to $f = 0.1$ (shown in Figure 9(c), 10(c) and 11(c)), $f = 0.9$ would lead to more coordinates/trajectory deviation (as shown in Figure 9(d), 10(d) and 11(d)). However, accurate count of objects (pedestrians) can be retained in the synthetic frames even if the flipping probability f is specified as 0.9 (small privacy bound). Thus, we can still use such synthetic videos to function specific application based on the count of objects, e.g., head counting and crowd density [23, 34]. To confirm such observation, we also detect and count all the pedestrians in each frame of the synthetic videos ($f = 0.1$ and $f = 0.9$).

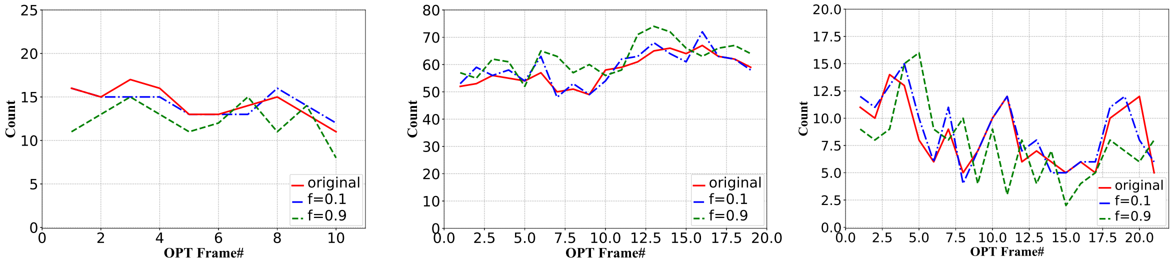
Figure 12 shows the pedestrian counts in the (optimized) key frames (after Phase I). The aggregated result lies very close to the original result when f is small. When f goes larger, the aggregated result is slightly more fluctuated, and more objects are generated in the frames. Figure 13 demonstrates the aggregated counts of pedestrians in each frame (after Phase II). Note that many objects (with the coordinates outside the frames; not between the "head" and "end") are suppressed in Phase II, making the object counts in different frames more accurate. Note that if multiple cameras capture more videos (e.g., surveillance or traffic monitoring cameras for the smart city) for joint analysis, the noise can be further cancelled in the applications.

6.4 Overheads

We evaluate the overheads of VERRO. Table 3 presents the runtime of the two phases and the required bandwidth for sending the synthetic videos to an untrusted recipient.

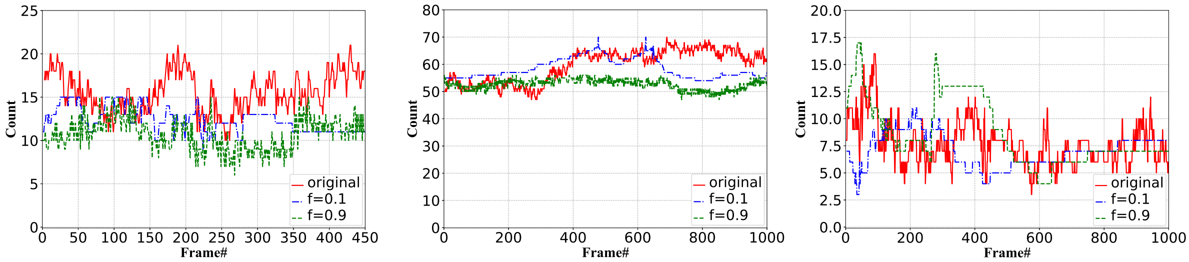
Table 3: Computational and Communication Overheads

Video	Phase I (Sec)	Phase II (Sec)	Bandwidth (MB)
MOT01	0.89	34.78	9.58
MOT03	1.56	36.12	16.6
MOT06	1.57	43.12	19.4



(a) MOT01 (b) MOT03 (c) MOT06

Figure 12: Object Counts in the Optimized Key Frames (by each frame)



(a) MOT01 (b) MOT03 (c) MOT06

Figure 13: Object Counts in the Synthetic Videos (by each frame)

The computational cost increases as the count of distinct objects increases (MOT01 has the least pedestrians while MOT06 has the most pedestrians). The results reflect a *sublinear* increase trend, which enables VERRO to be scaled to generate synthetic videos for longer videos (with more frames). In addition, although MOT06 has a lower resolution (less pixels) than MOT01 and MOT03, it is captured by a moving camera. Since more background scenes have to be interpolated, it requires longer runtime (but still efficient). Note that the runtime for object detecting and background scene(s) generation (1-2 minutes in our experiments) can be considered as computational costs for preprocessing.

Finally, the communication overhead for sharing three synthetic videos is almost identical to the original video size.

7 RELATED WORK

In the context of privacy preserving video publishing, many solutions have been proposed in literature (e.g., [6, 20, 41, 42, 44]). Saini et al. [41] have categorized such works in terms of the sensitive attributes obfuscated in the sanitization. These sensitive attributes include the evidence types *bodies, what*(activity), *where* (location where the video is recorded) and *when* (time when the video is recorded). In general, most of these works employ a *detect* and *blur* policy for only body attributes [6, 20, 42, 44] and some of them [15, 41, 47] aims at preserving the privacy against other three implicit inference channels.[3]

Specifically, these techniques often leverage computer vision techniques [20, 29] to first *detect* faces and/or other sensitive regions in the video frames and then *obscure* them. However, such *detect-and-protect* solutions have some limitations. For instance, the *detect-and-protect* techniques cannot formally quantify and bound the privacy leakage. In addition, blurred regions might still be reconstructed by deep learning methods [33, 37]. Last but not least, these techniques often use naive measures for quantifying the privacy loss in videos. For instance, in [20, 36], if faces are present, then it is considered as complete privacy loss, otherwise no privacy loss is reported. Fan [18] applied Laplace noise to randomly perturb the pixels in an image to ensure differential privacy for protecting specific regions of an image. However, the quality of the image is significantly deviated in the sanitized results. Our proposed privacy notion and the VERRO technique have addressed all the above limitations.

On the other hand, in the context of privacy preserving data publishing, the notion of differential privacy has emerged as a standard specification during past decade. This strong notion of privacy was first proposed by Dwork [13] to guarantee *indistinguishability* in the published data against an adversary armed with arbitrary background knowledge. Although differential privacy has been widely used to sanitize and release data in statistical databases [13], numeric data [45], location data [38], and search logs [25], to the best of our knowledge, no attempt has yet been made to benefit from differential privacy in video databases. Furthermore, to fully utilize differential privacy for sanitizing videos, we have defined our privacy notion based on a recently proposed locally implemented notion of differential privacy in which individuals in the videos (i.e., as objects) can directly interact with the sanitized result to ensure trustworthiness and fine-grained privacy. The emerging local differential privacy (LDP) models [4, 10, 16] have been utilized in a wide variety of applications (e.g., heavy hitters or histogram construction [4, 16], and frequent itemset mining [46]), but cannot be directly applicable to local video perturbation. VERRO complements the literature with strong privacy protection for (local) objects in the video against arbitrary background knowledge.

[3] The synthetic videos generated by VERRO can preserve the information of "where and when the videos are captured" while ensuring indistinguishability of objects (the linkage between every object and such inference channels can be broken to avoid leakage in the disclosure of the background scene).

8 CONCLUSION

Privacy concerns arise in considerable number of real world videos. To the best of our knowledge, we take the first cut to pursue indistinguishability for objects in the video by defining a novel privacy notion ϵ-*Object Indistinguishability*. We propose a two-phase video sanitization technique VERRO that locally perturbs all the objects in the video and generates a utility-driven synthetic video with indistinguishable objects, which can be directly shared to any untrusted recipient. In the synthetic videos, not only the object contents (e.g., different humans, and vehicle make/model/color), but also their moving trajectories in the video (e.g., a series of coordinates in different frames) can be effectively protected since every synthetic object and its trajectory can be possibly generated from any object in the original video. Experiments performed on real videos have validated the effectiveness and efficiency of VERRO. In the future, we will comprehensively study the utility of the synthetic videos in more application scenarios, and explore rigorous protection for objects which can be tracked in multiple videos.

9 ACKNOWLEDGEMENTS

This work is partially supported by the National Science Foundation (NSF) under awards CNS-1745894 and CNS-1564034, and the National Institutes of Health (NIH) under awards R01GM118574 and R35GM134927. The authors would like to thank the anonymous reviewers for their constructive comments.

REFERENCES

[1] 2012. YouTube~Official~Blog~2012
[2] 2019. https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm
[3] Bruno Abreu, Luis Botelho, and et.al. 2000. Video-based multi-agent traffic surveillance system. In *Intelligent Vehicles Symposium.* 457–462.
[4] Raef Bassily and Adam Smith. 2015. Local, private, efficient protocols for succinct histograms. In *Symposium on Theory of Computing.* 127–135.
[5] Vincent Bindschaedler, Reza Shokri, and Carl A Gunter. 2017. Plausible deniability for privacy-preserving data synthesis. *VLDB* (2017), 481–492.
[6] Michael Boyle, Christopher Edwards, and Saul Greenberg. 2000. The Effects of Filtered Video on Awareness and Privacy. In *CSCW.* 1–10.
[7] Yang Cao, Masatoshi Yoshikawa, Yonghui Xiao, and Li Xiong. 2017. Quantifying Differential Privacy under Temporal Correlations. In *ICDE.* 821–832.
[8] Paula Carrillo, Hari Kalva, and Spyros Magliveras. 2008. Compression independent object encryption for ensuring privacy in video surveillance. In *Multimedia and Expo.* 273–276.
[9] Rui Chen, Haoran Li, A Kai Qin, Shiva P. Kasiviswanathan, and Hongxia Jin. 2016. Private spatial data aggregation in the local setting. In *ICDE.* 289–300.
[10] Graham Cormode, Somesh Jha, Tejas Kulkarni, Ninghui Li, Divesh Srivastava, and Tianhao Wang. 2018. Privacy at scale: Local differential privacy in practice. In *Management of Data.* 1655–1658.
[11] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. 2004. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on image processing* 13, 9 (2004), 1200–1212.
[12] Ajay Divakaran, Regunathan Radhakrishnan, and Kadir A Peker. 2002. Motion activity-based extraction of key-frames from video shots. In *ICIP.* 932–935.
[13] C. Dwork. 2011. Differential privacy. *Encyclopedia of Cryptography and Security* (2011), 338–340.
[14] C. Dwork, A. Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science* (2014), 211–407.
[15] A. Erdelyi, T. Barat, P. Valet, T. Winkler, and B. Rinner. 2014. Adaptive cartooning for privacy protection in camera networks. In *AVSS.* 44–49.
[16] Ú. Erlingsson, V. Pihur, and A. Korolova. 2014. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *CCS.* 1054–1067.
[17] Fariba Fahroo and I Michael Ross. 2002. Direct trajectory optimization by a Chebyshev pseudospectral method. *Journal of Guidance, Control, and Dynamics* 25, 1 (2002), 160–166.
[18] L. Fan. 2018. Image Pixelization with Differential Privacy. In *DBSec.* 148–162.
[19] Huamin Feng, Wei Fang, Sen Liu, and Yong Fang. 2005. A new general framework for shot boundary detection and key-frame extraction. In *Multimedia information retrieval.* 121–126.
[20] Douglas A Fidaleo, Hoang-Anh Nguyen, and Mohan Trivedi. 2004. The networked sensor tapestry (NeST): a privacy enhanced software architecture for interactive analysis of data in video-sensor networks. In *Video surveillance & sensor networks Workshop.* 46–53.
[21] Elias Frentzos, Kostas Gratsias, Nikos Pelekis, and Yannis Theodoridis. 2007. Algorithms for nearest neighbor search on moving object trajectories. *Geoinformatica* 11, 2 (2007), 159–193.
[22] Feng Han, Ying Shan, Ryan Cekander, Harpreet S Sawhney, and Rakesh Kumar. 2006. A two-stage approach to people and vehicle detection with HOG-based SVM. In *Performance Metrics for Intelligent Systems.* 133–140.
[23] Marcus Handte, Muhammad Umer Iqbal, and et. al. 2014. Crowd Density Estimation for Public Transport Vehicles. In *EDBT/ICDT Workshops.* 315–322.
[24] Steven Hill, Zhimin Zhou, Lawrence Saul, and Hovav Shacham. 2016. On the (in) effectiveness of mosaicing and blurring as tools for document redaction. *Privacy Enhancing Technologies* 4 (2016), 403–417.
[25] Yuan Hong, Jaideep Vaidya, Haibing Lu, Panagiotis Karras, and Sanjay Goel. 2015. Collaborative Search Log Sanitization: Toward Differential Privacy and Boosted Utility. *IEEE Trans. Dependable Sec. Comput.* 12, 5 (2015), 504–518.
[26] Yuan Hong, Jaideep Vaidya, Haibing Lu, and Mingrui Wu. 2012. Differentially private search log sanitization with optimal output utility. In *15th International Conference on Extending Database Technology.* 50–61.
[27] Ravindran Kannan and Clyde L Monma. 1978. On the computational complexity of integer programming problems. In *Optimization and Operations Research.* 161–172.
[28] Kevin Karsch, Varsha Hedau, David A. Forsyth, and Derek Hoiem. 2011. Rendering synthetic objects into legacy photographs. *Trans. Graph.* (2011), 157.
[29] Takashi Koshimizu, Tomoji Toriyama, and Noboru Babaguchi. 2006. Factors on the sense of privacy in video surveillance. In *Continuous archival and retrival of personal experences.* 35–44.
[30] Sanjay K Kuanar, Rameswar Panda, and Ananda S Chowdhury. 2013. Video key frame extraction through dynamic Delaunay clustering with a structural constraint. *Visual Communication and Image Representation* (2013), 1212–1227.
[31] Xuan Li, Kunfeng Wang, Yonglin Tian, Lan Yan, Fang Deng, and Fei-Yue Wang. 2019. The ParallelEye Dataset: A Large Collection of Virtual Images for Traffic Vision Research. *Intelligent Transportation Systems* (2019), 2072–2084.
[32] Bingyu Liu, Shangyu Xie, Han Wang, Yuan Hong, Xuegang Ban, and Meisam Mohammady. 2019. VTDP: Privately Sanitizing Fine-grained Vehicle Trajectory Data with Boosted Utility. *IEEE Transactions on Dependable and Secure Computing* (2019), 1–1.
[33] Richard McPherson, Reza Shokri, and Vitaly Shmatikov. 2016. Defeating image obfuscation with deep learning. *arXiv preprint arXiv:1609.00408* (2016).
[34] Frank McSherry and Kunal Talwar. 2007. Mechanism Design via Differential Privacy. In *FOCS.* 94–103.
[35] A. Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. 2016. MOT16: A benchmark for multi-object tracking. *CoRR* (2016).
[36] S. Moncrieff, S. Venkatesh, and G. West. 2008. Dynamic privacy assessment in a smart house environment using multimodal sensing. *TOMM* (2008), 10.
[37] Seong Joon Oh, Rodrigo Benenson, Mario Fritz, and Bernt Schiele. 2016. Faceless person recognition: Privacy implications in social media. In *European Conference on Computer Vision.* 19–35.
[38] Lu Ou, Zheng Qin, Shaolin Liao, Yuan Hong, and Xiaohua Jia. 2018. Releasing Correlated Trajectories: Towards High Utility and Optimal Differential Privacy. *IEEE Transactions on Dependable and Secure Computing* (2018), 1–1.
[39] Lei Pan, Xiao-Jun Wu, and Yuan-Yuan You. 2005. Video shot segmentation and key frame extraction based on clustering. *Infrared and Laser Engineering* 34, 3 (2005), 341.
[40] Zhan Qin, Yin Yang, Ting Yu, Issa Khalil, Xiaokui Xiao, and Kui Ren. 2016. Heavy hitter estimation over set-valued data with local differential privacy. In *CCS.* ACM, 192–203.
[41] Mukesh Saini, Pradeep K. Atrey, Sharad Mehrotra, and Mohan Kankanhalli. 2014. W3-privacy: understanding what, when, and where inference channels in multi-camera surveillance video. *Multimedia Tools and Applications* 68, 1 (2014), 135–158. https://doi.org/10.1007/s11042-012-1207-9
[42] A. Senior, S. Pankanti, A. Hampapur, L. Brown, Ying-Li Tian, A. Ekin, J. Connell, Chiao Fe Shu, and M. Lu. 2005. Enabling video privacy through computer vision. *Security Privacy* (2005), 50–57.
[43] Alexander Toshev, Ameesh Makadia, and Kostas Daniilidis. 2009. Shape-based Object Recognition in Videos Using 3D Synthetic Object Models. In *CVPR.*
[44] M. Upmanyu, A. M. Namboodiri, K. Srinathan, and C. V. Jawahar. 2009. Efficient privacy preserving video surveillance. In *Computer Vision.* 1639–1646.
[45] Jaideep Vaidya, Basit Shafiq, Anirban Basu, and Yuan Hong. 2013. Differentially Private Naive Bayes Classification. In *2013 IEEE/WIC/ACM International Conferences on Web Intelligence.* 571–576.
[46] Tianhao Wang, Ninghui Li, and Somesh Jha. 2018. Locally differentially private frequent itemset mining. In *SP.* 127–143.
[47] T. Winkler and B. Rinner. 2013. Sensor-level security and privacy protection by embedding video content analysis. In *DSP.* 1–6.
[48] Nicolai Wojke and Alex Bewley. 2018. Deep cosine metric learning for person re-identification. In *WACV.* 748–756.
[49] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. 2017. Simple online and realtime tracking with a deep association metric. In *ICIP.* 3645–3649.
[50] Roman Yarovoy, Francesco Bonchi, Laks VS Lakshmanan, and Wendy Hui Wang. 2009. Anonymizing moving objects: How to hide a mob in a crowd?. In *Extending Database Technology.* 72–83.
[51] Qiang Zhu, Mei-Chen Yeh, Kwang-Ting Cheng, and Shai Avidan. 2006. Fast human detection using a cascade of histograms of oriented gradients. In *CVPR.* 1491–1498.