Privacy-Preserving Action Recognition using Coded Aperture Videos

Zihao W. Wang
Northwestern University
zwinswang@gmail.com

Vibhav Vineet
Microsoft Research
vibhav.vineet@microsoft.com

Francesco Pittaluga
University of Florida
f.pittaluga@ufl.edu

Sudipta Sinha
Microsoft Research
sudipsin@microsoft.com

Oliver Cossairt
Northwestern University
olivercossairt@gmail.com

Sing Bing Kang
Microsoft Research
sbkang@microsoft.com

Abstract

The risk of unauthorized remote access of streaming video from networked cameras underlines the need for stronger privacy safeguards. Towards this end, we simulate a lens-free coded aperture (CA) camera as an appearance encoder, i.e.*, the first layer of privacy protection. Our goal is human action recognition from coded aperture videos for which the coded aperture mask is* unknown *and does not require reconstruction. We insert a second layer of privacy protection by using non-invertible motion features based on phase correlation and log-polar transformation. Phase correlation encodes translation while the log polar transformation encodes in-plane rotation and scaling. We show the key property of the translation features being mask-invariant. This property allows us to simplify the training of classifiers by removing reliance on a specific mask design. Results based on a subset of the UCF and NTU datasets show the feasibility of our system.*

1. Introduction

Cameras as monitoring systems inside and outside the home or business is an important area of growth. However, as cameras that are connected online are prone to hacking, with images and videos illegally acquired potentially resulting in loss of privacy and breach of security.

In our work, we propose a novel privacy-preserving action recognition pipeline. This enhances the preservation of privacy from capture to executing visual tasks, as shown in Figure 1. By using a lensless coded aperture (CA) camera, which places only a coded mask in front of an image sensor, the resulting CA image would be visually unrecognizable and are difficult to restore with high fidelity. Decoding the image as a preprocessing step is an ill-posed inverse problem and requires expensive computation if the mask is non-separable. Instead, we extract motion features (translation, rotation, and scaling) using the Fourier-Mellin transform and use them as inputs to a deep neural network. We show that the translation features are invariant to the mask design, as long as its Fourier transform is broadband (*i.e.*, no zeros in the spectral magnitude). Specifically, the term "invariance" refers to the fact that the same translational features can be reproduced as long as the motion in the scene is identical. The translational features do not change over different mask patterns.

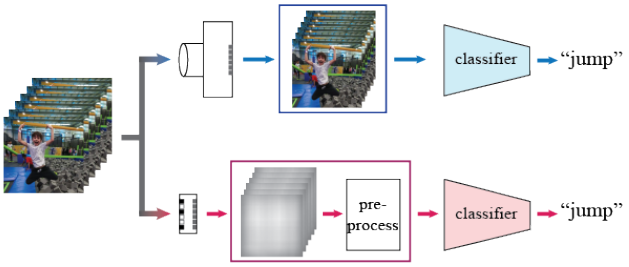

Figure 1: Comparison of imaging pipelines. Conventional vision systems (top row) may be vulnerable to loss of privacy due to hacking. Our simulated lensless coded aperture camera system (bottom row) has the benefit of preserving privacy while being able to classify human actions.

From a privacy point of view, the CA serves as the first layer of privacy protection, as CA images are visually incomprehensible. Our motion characterization serves as a second layer of privacy protection. The motion features are based on phase correlation between pairs of video frames, which whitens signal in Fourier space and only leaves motion signal.

The invariance property allows training to be done without regard to a specific mask design. Therefore, *we design a training mechanism which arbitrarily changes masks for each sample batch*. This training mechanism, along with the mask-invariant property, has practical meaning, as it makes commercial production of privacy preserving CA cameras

viable with random masks. In future, a camera manufacturer does not have to store the mask and share it with the third party who wants to develop recognition algorithms. Training data from one camera could be used to develop machine learning models that can be deployed on other cameras (with other masks).

2. Related work

Our work spans the areas of privacy-preserving vision systems, coded aperture imaging, motion feature extraction and action recognition. We briefly review representative approaches in these related fields.

2.1. Privacy-preserving approaches

Optics and imaging sensors. The rapid development of optics and sensors has brought emerging opportunities for privacy preservation. There are imaging sensors and/or modalities whose direct output is not visually recognizable. This matches the purpose of privacy preservation at optics/sensor level. An easy approach for preserving privacy is by defocusing [29]. Alternative optical solution is to put optical elements in front of sensors, *e.g*., cylindrical lens [27], diffraction gratings [36], or diffusers [2] in front of the sensor. Recovery of these images requires careful calibration of the imaging system and adequate computation.

Previous privacy preserving action recognition approaches include multiple extremely low resolution sensors [9], or compressive sensing (CS) [23]. More specifically, CS approaches require sequential multiple frame capture and a DMD array (which is costly and has fragile moving parts). Our approach only require one camera.

Firmware. There are also systems where the sensor firmware is modified to protect privacy before or during the sensing process. One approach is to embed privacy preserving schemes into the DSP architecture in cameras [8, 28, 42]. For example, in PrivacyCam [8], regions of interest are first identified based on background subtraction before being encrypted using AES. Other implementations involve embedding watermarks into the captured data [10, 22].

Vision algorithms. Software-based approaches for privacy preservation typically involve degrading or concealing information in images/videos. They include Gaussian blurring [6], face swapping [5], and scrambling [14]. Higher-level approaches use integrated classifiers [38] and/or recognizers [19] to allow sensitive information to be modified during capture and prior to sharing [43]. A recent technique improves privacy in the camera feed using an adversarial perturbation mechanism [31]. The adversarial learning mechanism has also been used for learning the optimal encoding schemes for different tasks [30].

2.2. Coded aperture imaging and photography

Coded aperture imaging was originally studied to fulfill the lack of imaging elements, namely, lenses and mirrors, in the field of astronomical X-ray and gamma-ray imaging in the 1960s [7, 12, 16]. Thereafter, the idea of extending pinhole cameras to cameras with masks consisting of designed patterns has been used for eliminating issues imposed by lenses and has found novel applications in extending depth-of-field [11, 13], extracting scene depth and light fields [25, 26, 40], and miniaturizing camera architectures [1, 4].

The blur effect caused by coded aperture can be used for privacy preserving applications. Li *et al*. has proposed coprime blurred pairs (CBP) for on/post capture video surveillance. A pair of coprime kernels can be used to blur images/videos. The original image/video can be recovered using the coprime kernels. CBP kernels are polynomial. This imposes higher numerical precision for the captured images. Compared to CBP, we focus our design on binary masks with significantly larger kernel sizes. Our goal is to perform action recognition without any form of restoration.

2.3. Motion features and action recognition

Finding motion features/descriptors from videos is a well-studied task in computer vision. *Global motion* can be used for camera calibration, image registration and video compression. The key idea is to find image-wise/block-wise matching between video frames via correlation [32] or image gradient [21]. Local motion features enable higher level vision tasks such as action recognition, anomaly detection, and video understanding. Early approaches include using handcrafted motion features, *e.g*., HOG/HOF [24] and dense trajectories [41]. Recent advances have utilized two-stream inputs (RGB + optical flow) [34] and 3D CNN [39] to learn spatio-temporal features [15].

State-of-the-art techniques for action recognition require both appearance video frames and optical flow features, as well as training on large-scale datasets, *e.g*., ImageNet and Kinetics. In our case, we explore the use of global motion features in the context of privacy preservation. In addition to serving as distinct signatures of basic actions, we show that they can be invariant to the coded aperture mask design.

3. Our algorithm

In this section, we describe how we compute features for action recognition *without* having to restore the images from a (simulated) lenless coded aperture camera. We call these features *TRS (translation, rotation, scale) features*. We first describe the image formation process.

3.1. Image formation

We consider a lens-free coded aperture imaging architecture, where a planar mask is placed in front of an imaging sensor. The encoding mask can be considered as an array of pinholes located at various lateral locations. The acquired image $\boldsymbol{d}$ can be numerically modeled as a convolution between the object image $\boldsymbol{o}$ and the point spread function (PSF) $\boldsymbol{a}$, i.e.,

$$\boldsymbol{d} = \boldsymbol{o} * \boldsymbol{a} + \boldsymbol{e}, \tag{1}$$

with $\boldsymbol{e}$ being noise. The convolution is applicable if the mask is far away enough from the sensor, such that each sensor pixel is able to see the entire mask pattern. If the mask-sensor distance is small (as in the case of FlatCam [4]), the mask design should consist of a smaller pattern replicated in a 2D array. The size of the smaller pattern should be such that each sensor pixel sees a version of it locally. This allows the output to be a result of convolution as well.

To restore the image, we convolve $\boldsymbol{d}$ with a decoding array $\boldsymbol{g}$ that satisfy the condition $\boldsymbol{g} * \boldsymbol{a} = \mathbf{I}$. This results in an estimate of the original image: $\hat{\boldsymbol{o}} = \boldsymbol{g} * \boldsymbol{d} = \boldsymbol{g} * (\boldsymbol{o} * \boldsymbol{a} + \boldsymbol{e}) = (\boldsymbol{g} * \boldsymbol{a}) * \boldsymbol{o} + \boldsymbol{g} * \boldsymbol{e} \approx \boldsymbol{o}$.

3.2. Translation based on phase correlation

Phase correlation was used first for global image registration [32] and then for motion/flow estimation [3, 18]. Compared to other motion estimation methods [37], phase correlation has the advantages of being computational efficient and invariant to illumination changes and moving shadows. We show how phase correlation can be used to characterize motion in coded aperture observations without knowing the mask design.

Assume there exists a translation between two video frames:

$$\boldsymbol{o}_1(\mathbf{p}) = \boldsymbol{o}_2(\mathbf{p} + \Delta\mathbf{p}), \tag{2}$$

where $\mathbf{p} = [x, y]^T$ and $\Delta\mathbf{p} = [\Delta x, \Delta y]^T$ are the spatial coordinates and displacement, respectively.

In frequency domain, translation gives rise to a phase shift:

$$\mathcal{O}_1(\nu) = \phi(\Delta\mathbf{p})\mathcal{O}_2(\nu), \tag{3}$$

where $\nu = [\xi, \eta]^T$ and $\phi(\Delta\mathbf{p}) = \exp^{i2\pi(\xi\Delta x + \eta\Delta y)}$. ξ and η are the frequency coordinates in Fourier space. By computing the cross-power spectrum and taking an inverse Fourier transform, the translation yields a delta signal:

$$\mathcal{C}_o(\xi, \eta) = \frac{\mathcal{O}_1^* \cdot \mathcal{O}_2}{|\mathcal{O}_1^* \cdot \mathcal{O}_2|} = \phi^* \frac{\mathcal{O}_2^* \cdot \mathcal{O}_2}{|\mathcal{O}_2^* \cdot \mathcal{O}_2|} = \phi(-\Delta\mathbf{p}), \tag{4}$$

$$\boldsymbol{c}(\mathbf{p}) = \delta(\mathbf{p} + \Delta\mathbf{p}). \tag{5}$$

The translation can be located by finding the peak signal; this feature is the basis of the original work [32], assuming a single global translation. Multiple translations result in an ensemble of delta functions. Note that these two equations are critical to our technique, as they show that computing translation is *independent of the coded aperture design*, as long as they have a broadband spectrum. Instead of finding the peak signal, we make full use of the computed image, treating it as a translation map (T-map).

3.3. Mask invariant property of T features

The convolutional transformation that generates a CA image encodes local motion in the original video to global motion in the resulting CA video. This makes the localization of the motion very challenging without restoration. However, we demonstrate that the global motions, such as translation, rotation, and scaling can still be retrieved using phase correlation. Following Eqs. (1) and (3), a translation relationship ($\Delta\mathbf{p}$) also exists:

$$\mathcal{D}_1(\nu) = \mathcal{O}_1 \cdot \mathcal{A} = \phi\mathcal{O}_2(\nu) \cdot \mathcal{A} = \phi\mathcal{D}_2(\nu), \tag{6}$$

where $\mathcal{A}$ denotes the Fourier spectrum of mask $\boldsymbol{a}$. The cross-power spectrum is then

$$\mathcal{C}_d(\nu) = \frac{\mathcal{D}_1^* \cdot \mathcal{D}_2}{|\mathcal{D}_1^* \cdot \mathcal{D}_2|} = \phi^* \frac{\mathcal{O}_2^* \cdot \mathcal{A}^* \cdot \mathcal{A} \cdot \mathcal{O}_2}{|\mathcal{O}_2^* \cdot \mathcal{A}^* \cdot \mathcal{A} \cdot \mathcal{O}_2|} \simeq \mathcal{C}_o. \tag{7}$$

Note that phase correlation has a magnitude normalization procedure while computing the cross-power spectrum. This step can effectively whiten the spectrum so as to eliminate global changes in appearance. This property provides an additional layer of privacy protection. In our implementation, we add a small number ϵ in the denominator of Eq. (7) to prevent division by zero. Regardless, the object spectrum will be unstable if $\mathcal{A}$ has near-zero elements.

We avoid the instability problem by randomizing the mask patterns, which substantially reduces the chance of zero values for $\mathcal{A}$. Examples are presented in Figure 2. With $\epsilon = 10^{-3}$, a pseudorandom mask (column 2 of Figure 2) shows near-identical T features compared to the T map computed from appearance frames directly. In Figure 2, the randomness decreased from mask 1 to mask 3. Mask 3 has the least amount of randomness and worst T feature degradation.

3.4. Mask design

We focus on 2D intensity binary mask patterns as they are more practical for implementation. As shown in Figure 2, the randomness in the mask pattern, which result in broadband spectra, preserves the T features compared to the T map computed from RGB frames. Figure 3 show representative masks that are considered. The pseudorandom mask (mask 1) provides a relatively uniform magnitude distribution. The separable mask (mask 2) based on maximum length sequence (MLS) have much stronger frequency response along the horizontal and vertical axes. Mask 3 is just

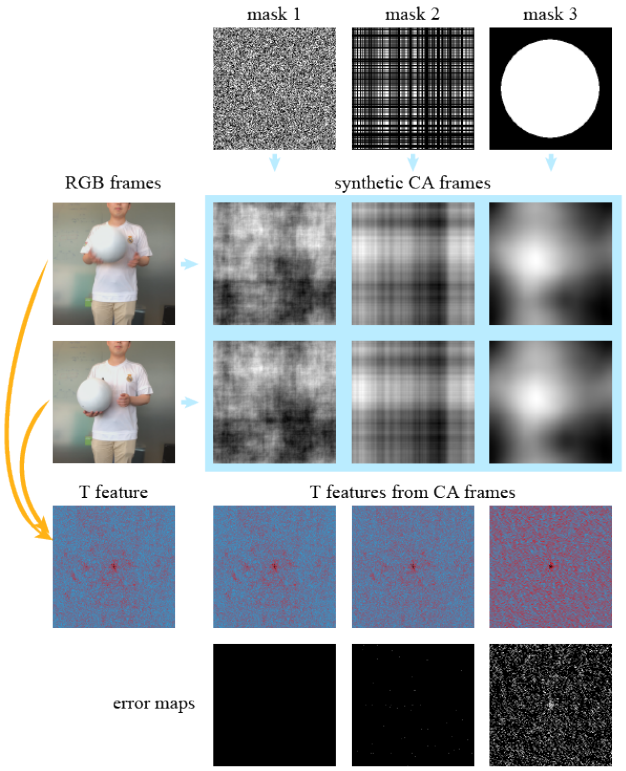

Figure 2: T features from different CA observations. Row 1: 3 different mask patterns (all 50% clear). Mask 1 is pixel-wise pseudorandom; mask 2 is a 2D separable Maximum Length Sequence (MLS). Rows 2 and 3: example RGB images and their corresponding synthetic CA frames. Row 4: T feature maps for the image pairs above them. Row 5: error maps, with the "ground truth" being the T map for RGB frames.

a round aperture, and it has undesirable dropoff at higher frequencies. We use pseudorandom masks for our experiments.

Note that since these masks are spatially as large as the image and non-separable in x and y (except row 1), high fidelity image restoration would be difficult and computationally-expensive [11]. We did not implement a restoration algorithm for these reasons.

We will show later that using only T features is less effective for action recognition (Figure 6). We investigate two extensions of the T features.

3.5. Extension 1: Translation-rotation-scale (TRS).

Given global translation, rotation, and scaling, we have $\boldsymbol{o}_1(\mathbf{p}) = \boldsymbol{o}_2(s\mathbf{R}\mathbf{p} + \Delta\mathbf{p})$, where s is a scaling factor and $\mathbf{R}$ is a rotation matrix with angle $\Delta\theta$. Translation $\Delta\mathbf{p}$ can be eliminated by taking the magnitude of the Fourier spectrum,

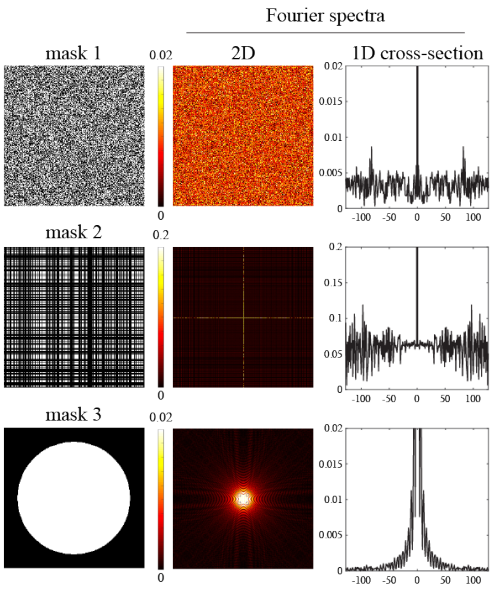

Figure 3: Two pseudorandom mask patterns and a round aperture (all 50% clear) and their Fourier spectra. This shows why pseudorandom patterns are desirable, since they retain high-frequencies.

i.e.,

$$|\mathcal{O}_1(\nu)| = |\mathcal{O}_2(s\mathbf{R}\nu)|. \tag{8}$$

If we treat the Fourier spectra as images and transform them into log-polar representations, i.e., $\mathbf{p} = [x, y]^T \Rightarrow \mathbf{q} = [\log(\rho), \theta]^T$, rotation and scaling become additive shifts on the two axes, i.e.,

$$|\mathcal{O}_1(\mathbf{q})| = |\mathcal{O}_2(\mathbf{q} + \Delta\mathbf{q})|. \tag{9}$$

This enables us to use phase correlation once again to locate rotation and scale. Qualitative examples are presented in Fig. 4.

3.6. Extension 2: Multi-stride TRS (MS-TRS).

We make a further extension to compute TRS features based on multiple strides in each video clip. This is to account for varying speeds of motion. For a video clip with length l, the TRS features in stride s are computed by:

$$T_i^{(s)}, RS_i^{(s)} = \mathcal{TRS}\{\boldsymbol{d}_{i\times s}, \boldsymbol{d}_{i\times s+s}\}, \tag{10}$$

where $i \in \{0, 1, ..., \lfloor \frac{l-s}{s} \rfloor + 1\}$ denotes all the possible consecutive indices within length l. For example, if a video clip of length 13 is given, the resulting $s2$ TRS features have 12 channels, 6 for T, and 6 for RS. In our case, we compare evaluation results for strides of 2, 3, 4, 6, with clip lengths of 13 and 19.

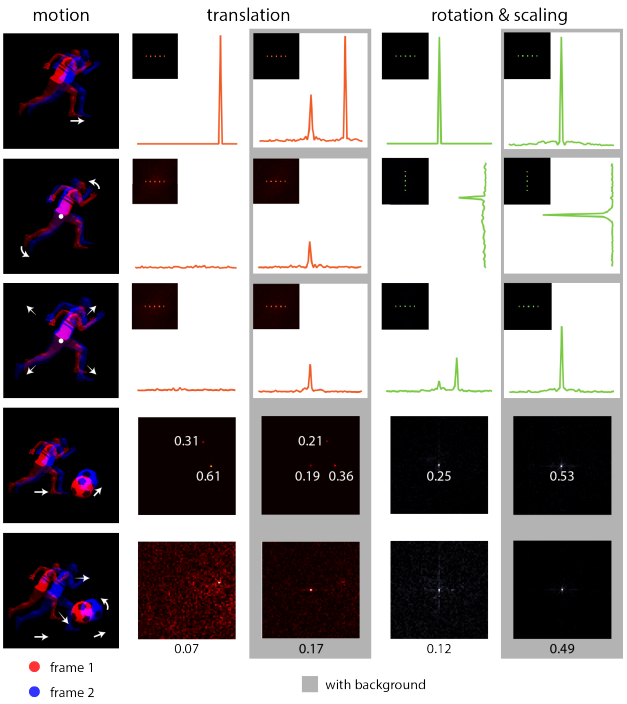

Figure 4: TRS feature comparison of synthetic motion cases. Row 1: pure translation, (0, 20) pixels. Row 2: pure rotation of $14°$, 10 pixels in RS y-axis. Row 3: pure scaling of $1.24\times$, 10 pixels in RS x-axis. Row 4: multiple translations. Highlighted are local peak values. Row 5: a combination of translation, rotation and scaling. Labeled are peak values. For the last two rows, zoom-in versions of the TRS maps are displayed.

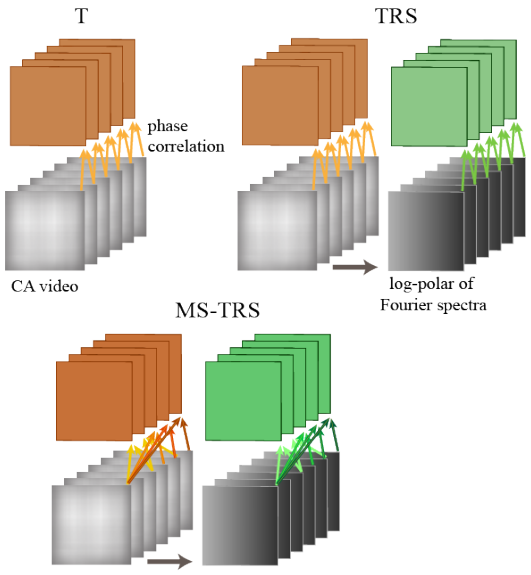

Figure 5: Visualization of proposed privacy-preserving features. T: translation only. TRS: translation, rotation, and scale. MS-TRS: translation, rotation, and scale under multiple strides.

4. Experimental results

In this section, we report results for the following experiments:

- Comparison with baselines: We compare the classification performance using regular and CA videos.
- Performance evaluation of our proposed T, TRS, and MS-TRS features;
- Comparison of effects using the same versus different or varying masks on training and validation;
- Comparison of using different MS-TRS configurations; this experiment is used to select an appropriate configuration for final testing.
- Testing of trained selected MS-TRS configuration.

We first describe the datasets and protocols used.

Datasets. We have evaluated our approach on the UCF-101 [35] and NTU [33] datasets. UCF-101 [35] contains 101 action classes with 13k videos. In our initial evaluation, we focus on indoor settings (more important from a privacy standpoint). Therefore, we created four different subsets from the 101 classes by selecting actions relevant to indoors (see Table 1). We also use the NTU [33] dataset which contains videos of indoor actions. We choose this dataset as it collects data using stationary cameras (we handle only static background for now). From our initial evaluation, we found that our proposed approach is better suited for more significant body motions. Because of this, we choose ten classes (with a mix of whole and partial body motions) for our final testing. Eight classes come from the NTU dataset and two classes are from the UCF dataset.

4.1. Protocol

Definitions. We use letters s and l to denote the stride and length of a video. For example, $s1, l4$ denotes four consecutive video frames. The number of input channels depends on the training mode.

Training and Validation. We use the first official train/test split from the UCF dataset and randomly select 20% of the training set for validation. Both the training and validation data is expanded using data augmentation to prevent overfitting. The data augmentation process is as follows.

- gray clips: Each video frame is loaded in as grayscale image at a resolution between 224 and 256. The aspect ratio is fixed at (240×320). The clip is then vertically

name	actions
UCF (5)	Writing on board, Wall pushups, blowing candles, pushups, mopping floor
UCF-body (09)	Hula hoop, mopping floor, baby crawling, body weight squat, jumping jack, wall push up, punch, push ups and lunges.
UCF-subtle (13)	Apply eye makeup, apply lipsticks, blow dry hair, blowing candles, brushing teeth, cutting in kitchen, mixing batter, typing, writing on board, hair cut, head assage, shaving beard, knitting.
UCF-indoor (22)	UCF-body and UCF-subtle

Table 1: Four different subsets of UCF-101[35] used in our evaluation. The number of classes is shown in brackets. Each class has about 10K video frames.

flipped with 50% chance. A $(224 \times 224 \times l)$ clip is then cropped and used as input.

- CA clips: Each CA clip first experiences the same augmentation step as gray clips. The CA simulation is computed at the resolution of 256×256 and rescaled back to 224×224. We simulate CA observations by computing element-wise multiplication in Fourier space between the Fourier transforms of the image and the mask kernel. We did not implement boundary effect for computation consideration. The diffraction effect is not accounted for as we observe minimal impact on the TRS features. Another reason is that simulating PSF for non-separable masks by matrix multiplication [11] is expensive.

- T features: The T features are generated from CA clips at the resolution of 256×256. The central 224×224 area is cropped as input. An l-frame CA clip results in $(l-1)$ T channels.

- TRS/MS-TRS features: In the TRS setting, the T features follow the same cropping. For RS, the R-axis uses center cropping while the S-axis is downsized to 224. An l-frame CA clip results in $2l$ channels, with l T channels and l RS channels stacked together. For MS-TRS, the resulting channels depend on the selected strides.

We use a batch size of 16 or 32. Each epoch, for both training and validation, prepares samples randomly from approximately 20% of all the possible frame combinations. 50 Epochs are used in our evaluation experiments. The percentage of accurate samples is reported. When reporting, we compute the running average accuracy of 5 epochs for better visualization.

Testing. During testing, we resampled each video at 3 spatial scales ($\mu \times \mu$ pixels, with $\mu = 224, 256, 300$) and 5 temporal starting frames evenly distributed across the video length. For example, using MS-TRS-$s346$-$l19$ configuration, a video with 100 frames will be used to generate five clips, starting at frames 1, 21, 41, 61, and 81, with each clip being 19 frames long. Each clip will be used to compute MS-TRS at three spatial scales. The final score for each video is computed by averaging the scores of the 15 clips.

Others. We use the VGG-16 CNN architecture, which contains approximately 134 million parameters. Adam optimizer is used with learning rate 0.0001, $\beta_1 = 0.9, \beta_2 = 0.999$. Since the CA observation is computed on-the-fly, we can change the underlying masks used in each batch. In this paper, we use "m1/m1" to refer to the setting where training and validation using the same fixed mask and "m1/m2" to refer to when training and validation uses two different masks. Finally, "dm1/dm2" denotes the setting where training and validation is done using variable masks. A pseudo-random binary mask is randomly generated for each batch. Note that the mask is fixed for all frames of a single video.

4.2. Initial evaluation

Baselines. Before training with our proposed privacy-preserving features (T, TRS, MS-TRS) as input, we first train one network on the original videos and three networks on the simulated CA videos as our four baselines. See the results in Table 2. The top-1 classification accuracy of 95% (row 1) for the original videos is our upper bound of what we can expect. After all, we do not expect our privacy preserving representation to add information. On the other hand, the performance of the baselines trained directly on CA videos (rows 2 to 4), will serve as our lower bounds. We expect our proposed features, which involve computation based on CA, to perform better than CA. The CA baselines show instability even when training and validation phases have the same mask. The network corresponding to the second row suffers from overfitting. Changing training masks for each batch does not improve the performance. These results show that it is difficult to train a network that can directly classify CA videos.

Variable masks during training. Our goal is to maximize the robustness of the designed features to the mask patterns. In order to achieve this, we change the training and validation masks by randomly generating a pseudo-random mask during each batch. We compare this dynamic training mechanism with two other modalities, *i.e.*, (1) training and validation using the same mask (m1/m1) and (2) training and validation using two different masks, no mask varia-

	training	validation
gray video	99.56 (99.86)	94.39 (95.91)
CA (m1/m1)	79.06 (92.65)	63.21 (86.96)
CA (m1/m2)	94.66 (95.17)	27.95 (40.55)
CA (dm1/dm2)	34.93 (36.61)	27.23 (36.96)

Table 2: Baseline comparison for UCF-05. Here, for the CA cases, training and validation are done directly on CA videos. The numbers are: average accuracy % of the last 5 epochs (maximum accuracy %). All clips have length 3.

tion during training (m1/m2). The results are presented in Figure 6.

For T features, the validation accuracy plateaus at about 60%. Even without training on dynamic masks, both validation accuracy (m1/m1 and m1/m2) gradually increase in a similar fashion. Dynamic training with variable masks does not improve the accuracy. This supports the fact that T features are invariant to the choice of masks.

For TRS features, using the same stride and length of the clips, the performance improves to around 70% for m1/m1. However, since the RS features are not mask-invariant, validation using a different mask does not have the same accuracy. Fortunately, by varying the masks during training, the performance stays the same as m1/m1. This is an interesting effect as, theoretically, the RS features do not have the same mask-invariant property. This drawback appears to be mitigated by changing the masks during training. This, in turn, enables us to test using an arbitrary mask.

For MS-TRS features, we observe a strong oscillation for the m1/m1 and a large gap between m1/m1 and m1/m2. The oscillation indicates the unstable progress for MS-TRS with the same training mask. The gap between m1/m1 and m1/m2 is very likely caused by the RS features, because RS is not mask-invariant. The use of dm1/dm2 overcomes these two drawbacks and achieves 77.8% validation accuracy.

Strides and clip length. In the case of TRS, we found that increasing the strides and clip lengths can improve the performance. The results are summarized in Table 3. In this case, the same mask was used during training and validation (m1/m1).

We evaluated different combinations of MS-TRS features. The training and validation for MS-TRS is under dm1/dm2 mode. The results are summarized in Table 4. For the same video length, using larger strides improves validation accuracy. For the same stride setting, *e.g.*, $s346$, processing more video frames improves performance. However, using longer stride and longer video, such as *i.e.* $s46, l19$, suffers from overfitting. The combination $s2346, l19$ is not evaluated as generating the 44-channel input on-the-fly becomes computationally expensive.

		training	validation
$ch6$	$s1$	97.23	82.33
	$s2$	97.44	82.49
	$s4$	98.66	85.16
$s2$	$ch4$	97.76	81.78
	$ch6$	97.44	82.49
	$ch10$	98.87	85.56

Table 3: Comparing performance of different strides and lengths of video, for TRS, m1/m1 on the UCF-05 dataset. The numbers are maximum accuracy percentages within the first 50 epochs. ch denotes the number of input channels.

	input shape	training	validation
$s2346, l13$	(224, 224, 30)	96.67	83.59
$s346, l13$	(224, 224, 18)	93.69	83.66
$s46, l13$	(224, 224, 10)	92.94	86.59
$s346, l19$	(224, 224, 26)	96.00	86.26
$s46, l19$	(224, 224, 14)	89.91	79.23

Table 4: Comparison of training and validation performances for MS-TRS, dm1/dm2 for UCF-05. Numbers are max accuracy percentage within the first 50 epochs.

	UCF-body	UCF-subtle	UCF-indoor
$s346, l13$	88.4 / 81.2	84.9 / 73.2	84.8 / 70.8
$s346, l19$	**90.5 / 83.4**	**86.1 / 76.4**	**88.6 / 72.8**
$s46, l13$	89.9 / 79.1	80.9 / 66.5	83.8 / 66.3

Table 5: Training and validation accuracies on different UCF subsets for networks trained on different MS-TRS configurations. UCF-body, UCF-subtle and UCF-indoor has 9, 13 and 22 classes respectively.

More action classes. We selected three MS-TRS settings from Table 4 and then trained networks for three larger datasets. These datasets are also subsets of UCF-101 actions focused on indoor settings and include body motions and subtle motions which primarily involve hand & face. The evaluation results are shown in Table 5.

4.3. Testing results

Based on the experiments on the UCF subset datasets, we selected *i.e.*, MS-TRS-$s346$-$l19$ as the best feature representation. Next, we computed MS-TRS-$s346$-$l19$ features on the 10-class combined dataset of NTU and UCF to examine the feasibility of our representation for daily activities. We used about one-sixth of the NTU videos for the eight classes for training to ensure we have a similar number of training examples as for the two UCF classes. In training phase, each class consists of 100 videos with more than 10K frames. We use a different data augmentation scheme

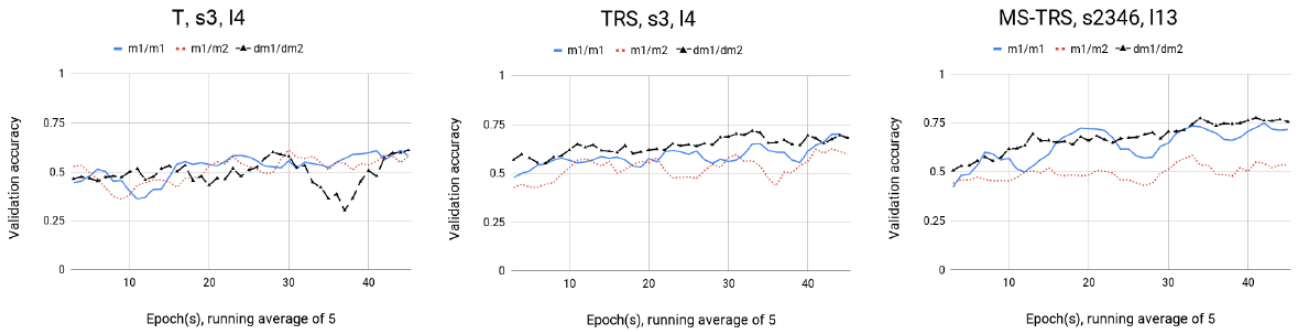

Figure 6: Comparison of validation accuracy for UCF-05, with training and validation: using the same mask (m1/m1), using two different masks (m1/m2), and based on a random mask per batch and a different random mask for validation (dm1/dm2). Note: s3 = stride of 3, s2346 = strides of 2, 3, 4, and 6.

ranking	class	top-1	top-2	top-3
1	hopping	97.1	100	100
2	staggering	94.3	97.1	100
3	jumping up	91.4	97.1	97.1
4	jumping jack †	81.1	91.9	100
5	body weight squats †	76.7	86.7	93.3
6	standing up	57.1	88.6	94.3
7	sitting down	51.4	82.9	100
8	throw	31.4	57.1	68.6
9	clapping	11.4	14.3	31.4
10	hand waving	5.70	14.3	20.0
	average	60.1	73.4	80.8

Table 6: Testing results for combined NTU and UCF 10 classes dataset. † indicates the class comes from UCF dataset, others are from NTU dataset. Ranking according to top-1 accuracy. The numbers are in percentages.

true class \ predicted class	1	2	3	4	5	6	7	8	9	10
1	97.1	0.0	2.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	94.3	0.0	0.0	0.0	0.0	2.9	2.9	0.0	0.0
3	0.0	8.6	91.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	10.8	2.7	81.1	5.4	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	3.3	20.0	76.7	0.0	0.0	0.0	0.0	0.0
6	0.0	28.6	8.6	0.0	0.0	57.1	5.7	0.0	0.0	0.0
7	0.0	37.1	5.7	0.0	0.0	5.7	51.4	0.0	0.0	0.0
8	2.9	51.4	2.9	0.0	0.0	0.0	11.4	31.4	0.0	0.0
9	0.0	65.6	0.0	0.0	0.0	0.0	15.6	6.3	12.5	0.0
10	0.0	31.4	2.9	0.0	0.0	0.0	42.9	8.6	8.6	5.7

Figure 7: Confusion matrix for the NTU-UCF-10 classes. The numbers are in percentages. Our technique works very well for large motion, but degrades as the motion become increasingly more subtle.

for the NTU dataset. Each NTU video is loaded at random height resolution between 460 and 520. The aspect ratio is fixed at $1080 : 1920 = 9 : 16$.

The central 240×320 region (same as the UCF classes) is cropped and used to compute CA and MS-TRS. For testing, each NTU video is loaded at 522×928 resolution. The central 256×256 video is cropped and used to compute CA and MS-TRS at different scales as described in the testing protocol. The overall top-1 accuracy is 60.1%. The top-1, 2, 3 accuracies for each class is reported in Table 6. The results indicate a large variation across classes. Our trained model is able to correctly recognize body motions such as hopping and staggering but is less accurate at differentiating between subtle hand motions such as clapping and hand waving. For further analysis, we show the confusion matrix in Figure 7. Interestingly, 58% of the misclassified samples are classified as "staggering".

5. Discussion

Restoration of coded aperture images. Restoration from CA images is possible but a non-trivial task. Deconvolution can be done if the mask design is known (including PSF or mask code, pixel pitch, distance between the SLM and the sensor). Approaches used include those of [4, 11] though their masks are separable in x and y whereas ours are not. Even when the mask and camera parameters are known, restoring our CA images can be expected to be substantially more computational expensive.

Our pseudorandom masks have approximately a delta function as their autocorrelation. Interestingly, this property enables object appearance recovery based on correlation-based methods. The autocorrelation of a CA image is equivalent to the autocorrelation of the scene image: $\boldsymbol{d} \star \boldsymbol{d} \simeq$

$(\boldsymbol{o} * \boldsymbol{a}) \star (\boldsymbol{o} * \boldsymbol{a}) = (\boldsymbol{o} \star \boldsymbol{o}) * (\boldsymbol{a} \star \boldsymbol{a}) \propto \boldsymbol{o} \star \boldsymbol{o}$. The object signal can, in principle, be recovered from its autocorrelation using a phase retrieval algorithm [17, 20]. However, such methods can only restore a coarse image (specifically, near binary quality at high contrast areas). More accurate image restoration is an interesting problem but is outside the scope of this paper.

Boundary effects in real CA images. Real CA images are expected to deviate from our simulated results. The most significant factor is the boundary effect. In our case, we simulate the CA observations by convolving the images and PSFs using FFT so as to achieve efficiency in training and validation. However, FFT-based convolution assume the signal is periodic, which is not the case for real cameras. A potential direction to ameliorate this problem is to zero-pad both image and mask, doubling both resolutions, apply FFT, element-wise multiply, and then crop to the original size after performing inverse FFT. This would generate simulated CA frames that are more consistent with ones captured with a real camera, but at a much higher computational cost.

6. Conclusions

There are several interesting takeaways from our experiments. First, training directly on the CA videos results in poor performance. Second, varying the mask at random during training reduces overfitting and improves performance. Third, using multiple strides with TRS (MS-TRS) as input works the best. This is likely attributed to its ability to adapt to different speeds of motion. Finally, results (for a subset of NTU and UCF mixed datasets) show that our technique works very well for large motion, but degrades as the motion become increasingly more subtle.

Additionally, the invariance property, theoretically, applies to both RGB videos and CA videos. As a result, in principle, this property should support transfer learning by first learning features from rich public RGB videos, with fine-tuning weights using actual CA videos. It would be interesting to investigate other visual tasks such as optical flow and anomaly detection. Other future directions include prototyping an actual CA camera system and collecting large-scale well-annotated CA datasets.

References

[1] J. K. Adams, V. Boominathan, B. W. Avants, D. G. Vercosa, F. Ye, R. G. Baraniuk, J. T. Robinson, and A. Veeraraghavan. Single-frame 3D fluorescence microscopy with ultraminiature lensless FlatScope. *Science Advances*, 3(12):e1701548, 2017.

[2] N. Antipa, G. Kuo, R. Heckel, B. Mildenhall, E. Bostan, R. Ng, and L. Waller. Diffusercam: lensless single-exposure 3d imaging. *Optica*, 5(1):1–9, 2018.

[3] V. Argyriou and T. Vlachos. A study of sub-pixel motion estimation using phase correlation. In *BMVC*, pages 387–396, 2006.

[4] M. S. Asif, A. Ayremlou, A. Sankaranarayanan, A. Veeraraghavan, and R. G. Baraniuk. Flatcam: Thin, lensless cameras using coded aperture and computation. *IEEE Transactions on Computational Imaging*, 3(3):384–397, 2017.

[5] D. Bitouk, N. Kumar, S. Dhillon, P. Belhumeur, and S. K. Nayar. Face swapping: Automatically replacing faces in photographs. In *ACM Transactions on Graphics (TOG)*, volume 27, page 39, 2008.

[6] M. Boyle, C. Edwards, and S. Greenberg. The effects of filtered video on awareness and privacy. In *ACM Conference on Computer Supported Cooperative Work*, pages 1–10, 2000.

[7] T. Cannon and E. Fenimore. Coded aperture imaging: Many holes make light work. *Optical Engineering*, 19(3):193283, 1980.

[8] A. Chattopadhyay and T. E. Boult. Privacycam: A privacy preserving camera using uCLinux on the Blackfin DSP. In *CVPR*, pages 1–8, 2007.

[9] J. Dai, J. Wu, B. Saghafi, J. Konrad, and P. Ishwar. Towards privacy-preserving activity recognition using extremely low temporal and spatial resolution cameras. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 68–76, 2015.

[10] L. De Strycker, P. Termont, J. Vandewege, J. Haitsma, A. Kalker, M. Maes, and G. Depovere. Implementation of a real-time digital watermarking process for broadcast monitoring on a TriMedia VLIW processor. *IEE Proceedings-Vision, Image and Signal Processing*, 147(4):371–376, 2000.

[11] M. J. DeWeert and B. P. Farm. Lensless coded-aperture imaging with separable Doubly-Toeplitz masks. *Optical Engineering*, 54(2):023102, 2015.

[12] R. Dicke. Scatter-hole cameras for x-rays and gamma rays. *The Astrophysical Journal*, 153:L101, 1968.

[13] E. R. Dowski and W. T. Cathey. Extended depth of field through wave-front coding. *Applied Optics*, 34(11):1859–1866, 1995.

[14] F. Dufaux and T. Ebrahimi. Scrambling for privacy protection in video surveillance systems. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(8):1168–1174, 2008.

[15] C. Feichtenhofer, A. Pinz, R. P. Wildes, and A. Zisserman. What have we learned from deep representations for action recognition? *Connections*, 19:29, 2018.

[16] E. E. Fenimore and T. M. Cannon. Coded aperture imaging with uniformly redundant arrays. *Applied Optics*, 17(3):337–347, 1978.

[17] J. R. Fienup. Phase retrieval algorithms: A comparison. *Applied Optics*, 21(15):2758–2769, 1982.

[18] H. T. Ho and R. Goecke. Optical flow estimation using Fourier Mellin transform. In *CVPR*, pages 1–8. IEEE, 2008.

[19] S. Jana, D. Molnar, A. Moshchuk, A. M. Dunn, B. Livshits, H. J. Wang, and E. Ofek. Enabling fine-grained permissions for augmented reality applications with recognizers. In *USENIX Security Symposium*, pages 415–430, 2013.

[20] O. Katz, P. Heidmann, M. Fink, and S. Gigan. Non-invasive single-shot imaging through scattering layers and around corners via speckle correlations. *Nature Photonics*, 8(10):784, 2014.

[21] Y. Keller and A. Averbuch. Fast gradient methods based on global motion estimation for video compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 13(4):300–309, 2003.

[22] E. Kougianos, S. P. Mohanty, and R. N. Mahapatra. Hardware assisted watermarking for multimedia. *Computers & Electrical Engineering*, 35(2):339–358, 2009.

[23] K. Kulkarni and P. Turaga. Reconstruction-free action inference from compressive imagers. *IEEE transactions on pattern analysis and machine intelligence*, 38(4):772–784, 2016.

[24] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, pages 1–8, 2008.

[25] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Transactions on Graphics (TOG)*, 26(3):70, 2007.

[26] C.-K. Liang, T.-H. Lin, B.-Y. Wong, C. Liu, and H. H. Chen. Programmable aperture photography: Multiplexed light field acquisition. In *ACM Transactions on Graphics (TOG)*, volume 27, page 55, 2008.

[27] S. Nakashima, Y. Kitazono, L. Zhang, and S. Serikawa. Development of privacy-preserving sensor for person detection. *Procedia-Social and Behavioral Sciences*, 2(1):213–217, 2010.

[28] P. Narayanan et al. The de-identification camera. In *Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*, pages 192–195, 2011.

[29] F. Pittaluga and S. J. Koppal. Pre-capture privacy for small vision sensors. *IEEE TPAMI*, 39(11):2215–2226, 2017.

[30] F. Pittaluga, S. J. Koppal, and A. Chakrabarti. Learning privacy preserving encodings through adversarial training. *arXiv preprint arXiv:1802.05214*, 2018.

[31] N. Raval, A. Machanavajjhala, and L. P. Cox. Protecting visual secrets using adversarial nets. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1329–1332, 2017.

[32] B. S. Reddy and B. N. Chatterji. An FFT-based technique for translation, rotation, and scale-invariant image registration. *IEEE Transactions on Image Processing*, 5(8):1266–1271, 1996.

[33] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. NTU RGB+D: A large scale dataset for 3D human activity analysis. In *CVPR*, June 2016.

[34] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014.

[35] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[36] L. Spinoulas, O. S. Cossairt, A. K. Katsaggelos, P. Gill, and D. G. Stork. Performance comparison of ultra-miniature diffraction gratings with lenses and zone plates. In *Computational Optical Sensing and Imaging*, pages CM3E–1. Optical Society of America, 2015.

[37] A. M. Tekalp. *Digital video processing*. Prentice Hall Press, 2015.

[38] R. Templeman, M. Korayem, D. J. Crandall, and A. Kapadia. Placeavoider: Steering first-person cameras away from sensitive spaces. In *Network and Distributed System Security Symposium (NDSS)*, pages 23–26, 2014.

[39] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497, 2015.

[40] A. Veeraraghavan, R. Raskar, A. Agrawal, A. Mohan, and J. Tumblin. Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing. *ACM Transactions on Graphics (TOG)*, 26(3):69, 2007.

[41] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *CVPR*, pages 3169–3176, 2011.

[42] T. Winkler and B. Rinner. Trustcam: Security and privacy-protection for an embedded smart camera based on trusted computing. In *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 593–600, 2010.

[43] R. Yonetani, V. N. Boddeti, K. M. Kitani, and Y. Sato. Privacy-preserving visual learning using doubly permuted homomorphic encryption. *arXiv preprint arXiv:1704.02203*, 2017.