A Prototype Sensor

The prototyping pipeline accepts a learned phase mask height map and culminates with fine-tuning of UtilityNet and AttackNet with the calibrated PSFs. The phase mask comprises of heights of individual pixels, which are discretized into steps of $200nm$. The mask is then fabricated using a Photonic Professional GT2, Nanoscribe GmbH (two-photon lithography 3D printer). We print using Nanoscribe's IP-Dip photoresist ($n = 1.5$) on a $700\mu m$ thick fused silica substrate. The printing is done using a $63\times$ Objective working in Dip-in Liquid Lithography mode (DiLL) mode. After printing, the substrate is laser cut to fit inside optics assembly of our lens. We also laser cut a black cardboard ring to act as our aperture and finally insert both the mask and aperture at the aperture plane of our lens.

Our prototype consists of a modular imaging lens and a sensor as shown in figure 1(B). To replicate the optics model of our simulation, we install the fabricated phase mask at the aperture plane of this lens. Edmund Optics fixed focal length modular lenses provide easy access to the aperture plane which makes it an ideal choice for our prototype. We use a $16mm$ Cx Series Fixed Focal Length Lens. To avoid fabricating a low resolution sensor, we use a high resolution sensor and downsample it. We use Lumenera's Lt545r camera which has a Sony IMX250 color imaging sensor. The imaging sensor has a native resolution of 2464×2056 pixels, a pixel size of 3.45μm and a 2/3" optical format. The sensor and lens combination achieves a field-of-view (FOV) of 30×25 degrees. Since our sensor simulation (sensor H in figure 2) hypothesizes a sensor with 16×16 pixels we intentionally down-sample our captured images from 2464×2056 pixels to 16×14 pixels and pad the vertical direction with reflective padding to end up with images of 16×16 pixels. If an actual low resolution sensor for the same imaging area, either the pixel pitch or the pixel size could be increased. Another scheme could be to implement pixel binning in the analog domain. The choice of a particular scheme would lead to different noise levels during capture but since they can be influenced by other factors as well, we simply assume a Gaussian noise level of $\sigma = 0.01$ for our simulations.

The learned phase mask has 2048×2048 simulated pixels with a pixel size of 4.25μm. The full size of the phase mask is 8.7mm, so we laser-cut an aperture of that size. We focus our sensor at 0.1m and calibrate our system by capturing a series of images of a point light source at 21 discrete depth values ranging from $1 - 10$m, which gives us the actual PSFs of our imaging system. These depths are the same ones for which we generate the simulated PSFs for learning the height map. At each depth, the exposure and gain on the camera for individual channels are adjusted to capture the most variation in PSFs. The captured PSFs are shown in figure 8. In order to account for non-idealities and misalignment in our system we fine-tune all downstream models with the actual (real) PSFs on the NYUv2 and VGGFace2 datasets. Further we collect a small dataset of 800 images and use 500 and 300 images for fine-tuning and testing of UtilityNet respectively. We divide the images carefully into different scenes such that images

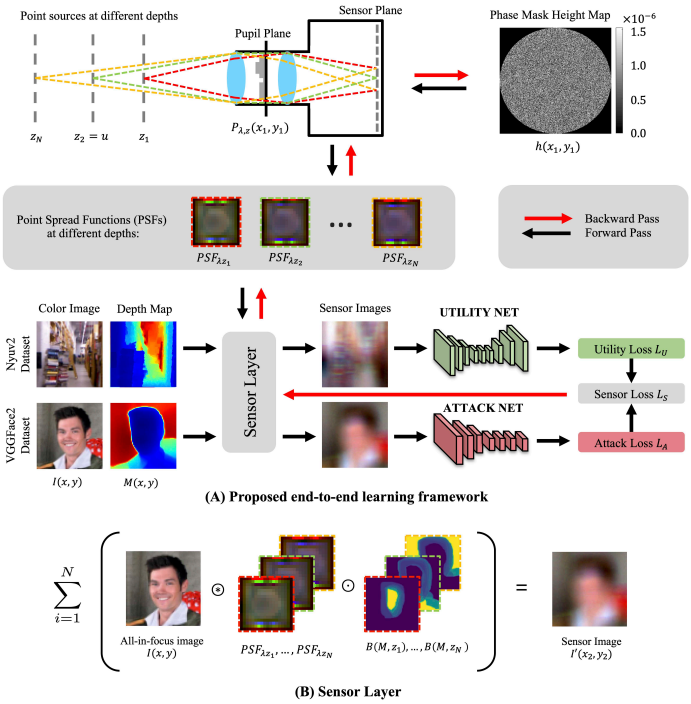

Fig. 7. Method. (A) Our proposed adversarial framework for end-to-end learning of sensor optics with respect to competing privacy and utility objectives. The optical sensor layer is simulated as modular lens system with a learnable phase mask. The sensor images generated by the sensor layer for the depth and face datasets are then used by UtitlityNet and AttackNet respectively. The adversarial sensor loss is back-propagated to update the phase mask height map. **(B)** Image formation model of the sensor layer where an all-in-focus image and it's corresponding depth map is taken as an input and the final sensor image is generated using a depth-dependent convolution.

from different labs, office and conference rooms belong to the fine-tune dataset while images from different lobbies belong to the test set to avoid over-fitting.

B Experimental Setup

B.1 Datasets

We utilize separate datasets for the utility and attack tasks. For the utility tasks, we use NYUv2 [33] for monocular depth estimation and NTURGBD120 [28] for action recogntion. For the attack tasks, we use VGGFace2 [5] for face identification and LFW [22] for face verification. Additionally, since our image formation model requires ground-truth depth maps to perform depth-dependent rendering of a scene, we generate pseudo-ground-truth depth maps for each image

in VGGFace2 and LFW using the pre-trained monocular depth estimation model from [43]. No such estimation was performed for NYUv2 and NTURGBD120 as each color image in these datasets comes paired with a ground-truth depth map captured by a Microsoft Kinect v1 and v2 respectively.

B.2 Evaluation Protocols

Depth prediction For depth estimation, all models output a predicted depth map of size 256×256 and are evaluated against a ground-truth depth map of size 256×256 regardless of the sensor resolution. We use threshold accuracy (δ_1) % of y_p s.t. $\max\left(\frac{y_p}{\hat{y}_p}, \frac{\hat{y}_p}{y_p}\right) = \delta_1 < 1.25$ where y_p and $\hat{y}_p$ denote a pixel in the ground truth and predicted depth maps, and n denotes the number of pixels in the depth maps.

Face identification and verification When evaluating face identification performance on VGGFace2, we follow the protocol described in [5]. That is, we first train a model for face recognition using the VGGFace2 training set consisting of face images of 8631 different subjects. Once training is complete, the classification layer of the model is removed and the remaining model is treated as a fixed feature extractor. The fixed feature extractor is then applied to the entire VGGFace2 evaluation set, which consists of 25,000 face images of 500 subjects, and a subset of the extracted features are used to train 1-vs-rest SVM classifiers for each subject. Finally, face identification performance is evaluated by applying the SVM on the remaining subset of the evaluation set.

When evaluating face verification performance on LFW, we follow the protocol described in [22]. Face verification differs from face identification in that the objective is not to identify a subject from a face image, but rather to determine whether two face images are of the same subject. As described in [22], we test our approach on 10 sets of 600 pairs of face images for 10-fold cross-validation. The feature extractor models used for face verification are trained on VGGFace2.

Our cropping protocol for face identification and verification differs slightly from convention in that bounding boxes are enlarged to enable deconvolution of encoded sensor images by AttackNet. Specifically, our cropping protocol consists of four steps: (i) Face bounding boxes (x_1, y_1, x_2, y_2) are estimated from the original uncropped face images using [65]; (ii) Uncropped face images are convolved with sensor PSFs to generate uncropped sensor images; (iii) Extended face bounding boxes $(x_1 - \frac{r}{2}, y_1 - \frac{r}{2}, x_2 + \frac{r}{2}, y_2 + \frac{r}{2})$, where r denotes the receptive field of the PSFs are calculated; (iv) Uncropped sensor images are cropped using the extended bounding boxes and fed to the respective downstream models.

B.3 Image Formation

For all simulations, we discretize depth maps using 21 depth values between 1m and 10m. Correspondingly, for each sensor, we generate 21 PSFs to convolve with scene points at the respective depths. For NYUv2 and NTURGD120 datasets we

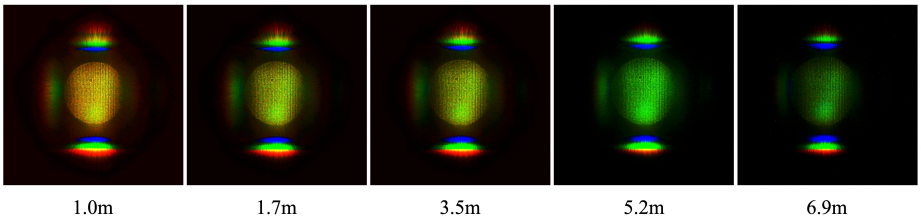

Fig. 8. Real PSFs from Prototype Sensor. Depth dependent calibration of the PSFs of our prototype sensor is performed by imaging a point light source at 21 different depths from $1 - 10m$. Here we show 5 of those PSFs. They vary significantly over different depths, which results in better depth estimation performance. Note, we show the PSFs here using our prototype sensor's native resolution of 2056×2464. When evaluating our prototype sensor, images are downsampled to 14×16 to match our simulations.

directly use the discretized versions of the provided depth maps for rendering. For the face images from VGGFace2 and LFW we perform additional pre-processing to render the face images at a range of distances from the sensor. This step is critical as a robust pre-capture privacy sensor should inhibit face identification for faces at any depth. The pre-processing consists of three steps. First, [65] is used to estimate a face bounding box and the larger side of the bounding box is taken as the face size. Second, a distance between 1m to 10m is randomly selected to position the face. Third, the image is resized such that the larger side of the face bounding box is the "right size" given the randomly selected depth and the specific sensor design. Since the actual sizes of the faces in VGGFace2 and LFW are not known, all faces are assumed to be 18.2cm, the average face size for American adults [66], when estimating what the "right size" should be.

B.4 Attack Models

We assess the efficacy of our learned sensor design at inhibiting facial identification attacks by four different attack models. For both the simulated and real privacy evaluations, we assume a white-box attack model, so attackers have access to the sensor design and can thus generate a large dataset consisting of triplets of source face images, corresponding simulated sensor images and ground-truth labels denoting the face ID. We generate such a dataset using the samples in VGGFace2 as the source face images and use this dataset to train four deep-learning-based attack models. For the first three attacks, we directly train classifier neural networks, EfficientNet-b0 [50], DenseNet121 [21], and ResNext101-32x8d [62], to recognize subjects in the gallery set of VGGFace2 from simulated sensor images. For the fourth attack, we train a U-Net [29] style neural network to reconstruct the original source images from the simulated sensor images and then train a classifier neural network (EfficientNet-b0) to recognize subjects in the gallery set of VGGFace2 from the reconstructions. Once trained, the final layer of the four classifiers is removed and the networks are used as feature extractors for

face verification on the LFW and prototype datasets for the simulated and real privacy evaluations respectively.

The UNet was trained using a combination of an L1 pixel loss and an L2 perceptual loss (as in [27, 13]) over the outputs of layers *relu1_1*, *relu2_2*, and *relu3_2* of VGG16 [46] pre-trained for image classification on the ImageNet [12] dataset:

$$\mathcal{L}_D = ||D(I') - I||_1 + \alpha \sum_{i=1}^{3} ||\phi_i(D(I')) - \phi_i(I)||_2^2, \tag{5}$$

where $I \in \mathbb{R}^{H \times W \times 3}$ denotes an all-in-focus image, $I' \in \mathbb{R}^{H \times W \times 3}$ denotes an encoded sensor image, $D : \mathbb{R}^{H \times W \times 3} \rightarrow \mathbb{R}^{H \times W \times 3}$ denotes a differentiable function representing the UNet, with learnable parameters, and $\phi_1 : \mathbb{R}^{H \times W \times 3} \rightarrow \mathbb{R}^{\frac{H}{2} \times \frac{W}{2} \times 64}$, $\phi_2 : \mathbb{R}^{H \times W \times 3} \rightarrow \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times 128}$, and $\phi_3 : \mathbb{R}^{H \times W \times 3} \rightarrow \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times 256}$ denote the layers *relu1_1*, *relu2_2*, and *relu3_2*, respectively, of the pre-trained VGG16 network. For optimization, we used the Adam optimizer [26] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon =$ 1e-8 and a learning rate of 1e-5, and trained the network until saturation.