Minimalist Vision with Freeform Pixels: Supplemental Material

Jeremy Klotz and Shree K. Nayar

Computer Science Department, Columbia University, New York NY, USA
{jklotz,nayar}@cs.columbia.edu

1 Importance of the Sensor Model

Here we examine the effect of training freeform pixels without including the sensor model in the network. Once trained, we evaluate the freeform pixels in a simulated minimalist camera that includes the optical effects and detector characteristics of the sensor model.

Using the synthetic dataset for counting patches described in Sec. 4 of the main paper, we generated 4 freeform pixels by training a minimalist camera without the sensor model. We then froze the learned masks and retrained the inference network with the sensor model included in the network. The parameters of the sensor model were chosen to be similar to that of our hardware prototype. After retraining the inference network, the 4 freeform pixels achieved 2.28 root-mean-square (RMS) error in the number of patches. By comparison, 4 freeform pixels that were trained for counting patches with the same sensor model incorporated in the network during training achieved 0.93 RMS error in the number of patches. This performance gap between the two minimalist cameras demonstrates that including the sensor model in the network during the training process is critical to generate performant freeform pixels.

2 Camera Architecture Details

Table 1 lists the components used in our prototype minimalist camera. Each detector is connected to a transimpedance amplifier with a gain of 10^7 V/A. In the lightweight vision experiments, we used a National Instruments USB-6363 data acquisition unit to simultaneously read out the freeform pixel measurements and trigger the training camera. Since the detectors and training camera are sensitive to near-infrared wavelengths, we mounted a filter in front of the minimalist camera to block near-infrared light.

The sensor model parameters corresponding to the hardware prototype were either empirically measured or extracted from component datasheets. First, the detector datasheet [1] specifies the active area to be $0.88 \times 0.88\,\mathrm{mm}^2$ and publishes the directional response. We use $\sigma_r = 400\,\mu\mathrm{V}$ as the standard deviation of the read noise. Quantization noise and sensor saturation are based on a 16-bit detector that saturates at 3.2 V. Finally, our process of printing masks on a transparency can only fabricate masks with transmittance values in the range

Table 1: List of components in the prototype minimalist camera.

Component	Quantity	Description
Detector	24	Hamamatsu S9119-01
Amplifier	24	TLV521DCKR
Multiplexer	1	ADG732BCPZ
Microcontroller	1	STM32WB5MMG
Photovoltaic	4	PowerFilm MP3-37
Supercapacitor	8	11 mF, each
Training Camera	1	Basler daA1920-160uc
Training Camera Lens	1	Edmund Optics 3 mm, $f/2.5$
Infrared Filter	1	Schott KG3

$0.01 \leq M(x, y) \leq 0.67$. We account for this fabrication limitation by scaling the mask transmittance values to this range during the training process.

3 Lightweight Vision Experimental Details

Slight mismatches between the sensor model and hardware prototype cause deviations between the simulated and real measurements of each freeform pixel. Furthermore, radiometric and geometric misalignments between the freeform pixels and training camera contribute to this mismatch. To account for this mismatch after the freeform pixels are fabricated, we retrain the inference network using pairs of real measurements generated by the prototype and their corresponding ground truth labels. This processes necessitates the capture of two datasets for each lightweight vision experiment. The first dataset contains a training video that is only used to the generate masks of the freeform pixels that will be fabricated. Once the masks are fabricated, a dataset is captured containing a video from the training camera and corresponding measurements from the freeform pixels. This dataset, which is summarized in Tab. 2 for each task, is used to retrain the inference network of the hardware prototype and train simulated minimalist cameras and baseline cameras.

3.1 Workspace Monitoring

The networks for counting people were trained by minimizing the mean squared error between the predicted and ground truth people count. The networks for the remaining tasks (detecting the state of the door and occupancy of the zones) were trained by minimizing the cross-entropy loss. At test time, the predicted people count from both the baseline and minimalist cameras is rounded to the nearest integer and then filtered using a 2-second median filter. In the supplemental video, the outputs produced by the minimalist camera for detecting the state of the door and the zone occupancy are filtered using a 0.5-second median filter.

Table 2: Sizes of the datasets used in the lightweight vision experiments.

Experiment	Dataset Split	Duration (min.)	# Samples
Workspace Monitoring	Training	38	68,069
	Validation	11	19,400
	Testing	10	18,720
Lighting Estimation	Training	17	31,411
	Validation	6	10,162
	Testing	6	10,738
Traffic Monitoring	Training	166	23,951
	Validation	21	3,479
	Testing	42	7,118

We fabricated 16 freeform pixels for counting people. We then used a greedy algorithm to iteratively remove the least important pixels from the collection to evaluate the counting performance using a smaller number of pixels. At each iteration in this algorithm, the "least important" freeform pixel is the one which, when removed from the collection, admits the smallest increase in validation loss. Figure 7(d) in the main paper shows the performance of subsets of the 16 freeform pixels obtained using this approach.

As explained in the main paper, we generated a dataset to evaluate the face identification performance of the 16 freeform pixels designed for counting people. We constructed the dataset using 100 randomly chosen identities in the CelebA dataset [2] that each appear in at least 20 images. The training, validation, and testing sets are composed such that each set contain images of all 100 individuals. We trained minimalist camera networks to convergence by performing a grid search over the batch size, learning rate, and the inference network's width and depth.

3.2 Traffic Monitoring

Both the minimalist camera and baseline camera use the temporal history of measurements over a period of one second (a stack of 30 measurements) to estimate the average traffic speeds. We apply forward differencing in the time domain to the measurement stack before passing it through the inference network. We found empirically that applying forward differencing improved the performance of both the baseline and minimalist camera networks.

The validation and test sets for traffic monitoring are extracted by randomly sampling five-minute clips from the eight-hour video, as described in the main paper. The remaining portions of the video are used for training. The datasets are generated by extracting overlapping one-second periods from the video. Some clips that do not contain any traffic are removed from the datasets. To retrain the inference network of the hardware prototype, we generated a larger number of training and validation samples (142,933 and 20,770, respectively) by extracting one-second periods with more aggressive overlap.

At test time, the estimated traffic speeds from both the minimalist camera and baseline camera are filtered using a 2-second median filter. We also observed that the object detector used for ground truth labeling is only accurate within a field-of-view that is slightly smaller than that of the baseline and minimalist camera. This caused labeling errors when a moving vehicle appears near the edge of the image. To minimize the effect of this labeling error on the computed performance (i.e. the RMS error of the predicted traffic speeds), we set the predicted speed of both the minimalist and baseline cameras to 0 when the ground truth speed is less than 3 miles per hour.

References

1. Hamamatsu Photonics: S9119-01 Photodiode Datasheet
2. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep Learning Face Attributes in the Wild. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (Dec 2015)