Supplementary Material for Booster-SHOT: Boosting Stacked Homography Transformations for Multiview Pedestrian Detection with Attention

Jinwoo Hwangnum
Seoul National University
luorix@snu.ac.kr

Philipp Benz
Deeping Source Inc.
philipp.benz@deepingsource.io

Pete Kim
Deeping Source Inc.
pete.kim@deepingsource.io

Additional to our main paper, we provide supplementary material. The following results explore the generalization capability of BoosterSHOT, the performance of HAM compared to other attention mechanisms, and ablations related to auxiliary losses and distance between homography planes.

1. Preliminaries

For a pixel in image I^i, with pixel coordinates (u, v) and its corresponding position in the 3D space (X, Y, Z), we can write the following equation using the pinhole camera model:

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \lambda \mathbf{G}^i [\mathbf{R}^i | \mathbf{t}^i] \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. \tag{1}$$

Here, λ is a scaling factor that accounts for possible mismatches between image and real 3D space increments. The above equation can be written as follows for the ground plane ($Z = 0$):

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{bmatrix} \theta^i_{11} & \theta^i_{12} & \theta^i_{13} & \theta^i_{14} \\ \theta^i_{21} & \theta^i_{22} & \theta^i_{23} & \theta^i_{24} \\ \theta^i_{31} & \theta^i_{32} & \theta^i_{33} & \theta^i_{34} \end{bmatrix} \tag{2}$$

$$\begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix} = \begin{bmatrix} \theta^i_{11} & \theta^i_{12} & \theta^i_{14} \\ \theta^i_{21} & \theta^i_{22} & \theta^i_{24} \\ \theta^i_{31} & \theta^i_{32} & \theta^i_{34} \end{bmatrix} \tag{3}$$

$$\begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} = \mathbf{\Theta}^i \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}. \tag{4}$$

We can apply the inverse of $\mathbf{\Theta}^i$ to both sides of the equation and multiply by $\mathbf{F}^i$ to obtain a matrix mapping from image coordinates directly to the ground plane grid. That matrix can be written as

$$\mathbf{H^i} = \mathbf{F}^i (\mathbf{\Theta}^i)^{-1}, \tag{5}$$

which is a homography matrix.
To expand this approach to planes other than the ground plane ($Z \neq 0$), we adopt SHOT's [1] method and replace $\mathbf{\Theta}^i$ with

$$\mathbf{\Theta}^i = \begin{bmatrix} \theta^i_{11} & \theta^i_{12} & \theta^i_{14} + k\Delta z \theta^i_{13} \\ \theta^i_{21} & \theta^i_{22} & \theta^i_{24} + k\Delta z \theta^i_{23} \\ \theta^i_{31} & \theta^i_{32} & \theta^i_{34} + k\Delta z \theta^i_{33} \end{bmatrix} \tag{6}$$

where θ^i_{j3} ($j \in \{1, 2, 3\}$) are the values omitted in Equation 3, Δz is the distance between homographies, and k is any non-negative integer lower than the total number of homographies (thus denoting all possible heights for the homography). With this new $\mathbf{\Theta}^i$, we can retain the homography matrix representation shown in Equation 5.

2. Supplementary Analysis

BoosterSHOT performance under various settings for distance between homography planes In the main manuscript, we visualized results for BoosterSHOT with 60cm between each homography plane. To explore whether the distance between homography planes has any significant impact on the performance of BoosterSHOT, we performed additional experiments with the distance between homography planes set to 20cm and 40cm, respectively. As shown in Table 1, variations of BoosterSHOT showed comparable performance. Variations with more distance between homography planes even showed a consistent 0.3 increase in MODA while other metrics remained comparable to or better than MVDeTr.

Spatial gate results under various settings for distance between homography planes We further investigate the relation between our spatial attention heatmaps and the distance between homography planes. To this end, we provide Figure 1 showing 4 spatial attention heatmaps. The first row shows spatial attention heatmaps for a homography plane at 40cm above the ground and the second row shows heatmaps for a homography plane at 120cm above

Table 1. BoosterSHOT performance based on distance between homography planes

	MODA	MODP	precision	recall
Booster-SHOT (Δz=0.2)	**94.4**	92.4	99.0	95.3
Booster-SHOT (Δz=0.4)	**94.4**	92.4	99.0	95.3
Booster-SHOT (Δz=0.6)	**94.4**	**92.5**	98.2	**96.2**
Booster-SHOT	94.1	91.7	98.3	95.7
MVDeTr	93.7	91.3	**99.5**	94.2

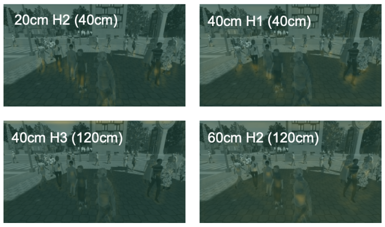

Figure 1. Comparison of spatial attention maps for homography planes of equal height

the ground. Heatmaps corresponding to the same height attend to similar regions of the image, while those at different heights show different patterns when it comes to the highlighted regions. This is an indication of the validity of our claim that spatial attention is reliant on and consistent with the height of each of the homography planes.

Effect of per-view loss on model performance To quantify the effect of the per-view loss on BoosterSHOT, we experimented with 3 different settings. Per-view loss indicates any loss used only during training, such as losses for foot position regression in each camera view. We show the results for BoosterSHOT with no per-view losses, with a loss term for foot position regression, and with loss terms for both foot and head position regression in Table 2. For the pre-existing literature, MVDet used both head and foot loss, while SHOT and MVDeTr used only foot regression as a subtask during training. Introducing a head position regression subtask during the training phase shows a non-negligible increase in performance of 0.7% in terms of MODA when compared with our default approach using only foot regression. Removing all per-view losses resulted in a 0.3% MODA decrease.

Table 2. BoosterSHOT (deformable transformer) performance by per-view loss

BoosterSHOT + Tr	MODA	MODP	precision	recall
no loss	93.8	92.0	98.3	95.4
foot (default)	94.1	91.7	98.3	95.7
head + foot	**94.8**	**92.1**	**98.4**	**96.3**

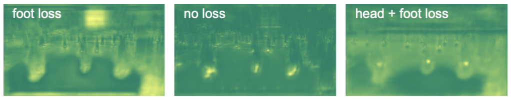

Figure 2. Comparison of image features with and without per-view loss

In Figure 2, we compare image features extracted from BoosterSHOT's feature extractor with different kinds of auxiliary loss settings. The features extracted when trained with no additional losses are more concentrated, while those from extractors trained with either foot regression loss or both foot and head regression losses are shown to be more broad. In addition, comparing the head + foot loss and foot loss feature heatmaps shows that the foot loss induces a bias toward the ground plane on which feet are placed, whereas the additional head loss counters that and helps the features attend overall to the entire body of the pedestrians. Table 2 shows that including the head regression auxiliary loss provides an additional increase in performance compared to only using the foot regression auxiliary loss.

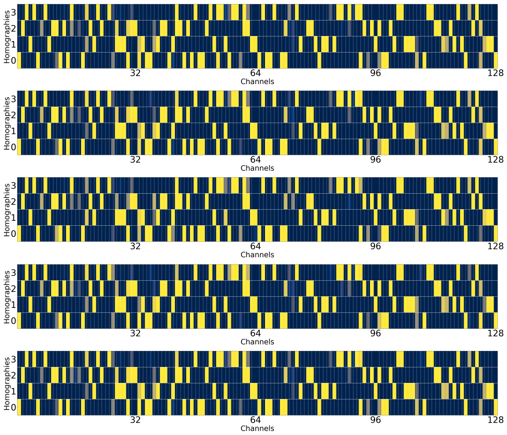

Figure 3. Channel selection heatmap (Camera 2 to 6 from top to bottom)

Additional Results We include heatmaps showing which channels were selected for each homography in each camera view. Figure 6 in the main manuscript showed the selection heatmap for Camera 1 in MultiviewX, while Figure 3 shows the selection heatmaps for Cameras 2 through 6. Consistent with the previous findings, the channels that are selected for the majority of the test set for each homography are shown to be consistent across camera views.

Computational cost, memory consumption and runtime. We evaluate the benefits of our method in terms of computational cost, via Giga FLoating-point OPerations (GFLOPs), memory consumption, and runtime in seconds. Through one-time inference, we account for floating-point operations such as addition, multiplication, and division.[1] For evaluating memory consumption, we count model parameters and buffers. For the runtime, we ran 20 randomly generated tensors with values between 0 and 1 through the models.

With our method, upon selecting K channels from the image heatmaps for each homography and using D homographies, the input to the bird's-eye-view (BEV) heatmap generator has $K \times D$ channels. A smaller K indicates fewer channels in the input and fewer parameters in the BEV heatmap generator. In addition, as the spatial attention module input in our spatial gate has K channels, we further save on computations during channel-wise pooling.

Table 3. Computational complexity comparison between methods for 4 homographies

Method	GFLOPs	# of parameters	runtime (sec)
SHOT	4.71k	19.0M	0.33 ± 0.076
SHOT + HAM (top 4)	4.09k	14.9M	0.29 ± 0.071
SHOT + HAM (top 16)	4.23k	16.4M	0.28 ± 0.074
SHOT + HAM (top 32)	4.42k	18.5M	0.22 ± 0.054
MVDeTr	2.59k	12.8M	0.19 ± 0.00032
MVDeTr + HAM	2.69k	12.9M	0.21 ± 0.0029
Booster-SHOT	2.54k	13.3M	0.19 ± 0.0015
Booster-SHOT w/ Tr	2.54k	12.9M	0.22 ± 0.0016

Utilizing the PTFLOPS[2] package, we count the number of FLOPs for SHOT and SHOT with HAM. Table 3 shows results for SHOT and SHOT with HAM using 4, 16, and 32 channels per homography. We find that SHOT with 4 homographies, our baseline, takes up 4705.73 GFLOPs and 19048768 parameters. SHOT with HAM and 4 homographies using only the top-32 channels for each homography takes up 4423.78 GFLOPs and 18530184 parameters, yielding 6.16% improvement in computational cost and 2.63% improvement in memory usage while outperforming SHOT in all four metrics. Our best performing approach (top 16 in Table 3) takes up 4228.66 GFLOPs and 16438088 parameters, outperforming SHOT in all four metrics while achieving a 13.7% and 10.2% reduction in computational cost and memory usage. Our most lightweight approach (top 4 in Table 3) takes up 4089.63 GFLOPs and 14881112 parameters, showing a 21.6% and 13.2% reduction, respectively. It also outperforms SHOT in MODA, precision, and recall with comparable MODP (-0.2%) (see Table 5). In addition, we also tested the additional computational cost and runtime incurred by applying our HAM to previous approaches such as MVDet, SHOT, and MVDeTr. We test the computational cost and runtime of pre-existing methods before and after applying HAM. All experiments use an input tensor of size (1, 7, 3, 720, 1280) (consistent with Wildtrack/MultiviewX).

For SHOT, because we replaced the "soft selection module" with the lighter HAM (HAM takes fewer channels as input, reducing the number of parameters), computational cost decreases by 5.99%. We also found that by reducing the number of channels (K) selected per homography, applying HAM to SHOT can reduce the computational cost by 13.2% while still improving the performance. For MVDeTr, adding HAM (one homography) results in a 3.51% increase in computational cost. The additional cost of increasing the number of homographies is also minimal: SHOT + HAM (6 homographies): 4444.46 GFLOPs SHOT + HAM (8 homographies): 4480.92 GFLOPs. The results of the runtime evaluation overall follow the same trend as that of the computational cost.

Overall, the results indicate that HAM incurs minimal additional cost when naively applied to existing methods and enables to tune the model to reduce computational memory costs while boosting performance.

3. Ablation Experiments

Number of homographies As shown in SHOT [1], as using multiple homographies is essentially a quantized version of a 3D projection, using more homographies leads to better performance for multi-view pedestrian detection. As our method assigns fewer channels to each homography as the number of homographies increases, we test the performance of SHOT with our module implemented for 2, 4, 6, and 8 homographies. Overall, all four metrics show improvement as the number of homographies increases (see Table 4). The 6 homography case has the highest MODP and recall while the 8 homography case has the highest precision. Both cases mentioned above have the highest MODA. As the overall performance is very similar, we conclude that the improvement from the increased number of homographies has reached an equilibrium with the decreased number of channels passed to each homography.

Table 4. Performance depending on the number of homographies

Method	#H	MultiviewX			
		MODA	MODP	precision	recall
SHOT	5	88.3	82.0	96.6	91.5
SHOT + HAM	2	89.4	80.8	95.2	94.2
SHOT + HAM	4	90.6	82.2	96.8	93.8
SHOT + HAM	6	**91.4**	**83.1**	97.4	**93.9**
SHOT + HAM	8	**91.4**	82.6	**97.5**	93.8

Number of top-K channels Our approach initially determined the number of channels selected per homography

[1] As the post-processing cost is unchanged regardless of our settings, we take into account the calculations required to go from the RGB image input to a bird's-eye-view heatmap

[2] https://github.com/sovrasov/flops-counter.pytorch

based on the number of homographies and the number of input channels. For example, our base approach for 128 input channels and 4 homographies involves selecting the top-32 channels for each homography. We further test the performance of our module when we fix the number of channels selected per homography (hereon denoted as K in accordance with the name top-K selection) and change the number of output channels accordingly. Setting $K = 64$ for 4 homographies and 128 input channels indicates we take the top-64 channels for each homography and output $64 \times 4 = 256$ channels. Table 5 outlines the results we get for $K = 4, 8, 16, 32, 64, 128$. For MODA, MODP and precision, using the top-16 channels for each homography outperforms the other instances with considerable margins. The top-32 instance (our base approach) improves on the top-16 instance only for recall. We conclude that our channel selection approach is effective in removing irrelevant channels and concentrating relevant information into selected channels for each homography.

Table 5. Performance depending on the number of selected channels

Method	K	MultiviewX			
		MODA	MODP	precision	recall
SHOT + HAM	4	90.6	81.8	97.7	92.7
SHOT + HAM	8	90.4	82.2	97.9	92.4
SHOT + HAM	16	**91.8**	**82.6**	**98.9**	92.9
SHOT + HAM	32	90.6	82.2	96.8	**93.8**
SHOT + HAM	64	90.2	82.2	96.9	93.2
SHOT + HAM	128	89.2	81.8	96.0	93.0

Attention Mechanisms In Table 6, we outline the effects of the channel gate and the spatial gate on MVDet, as well as their combination (HAM). It can be observed that both the channel gate and the spatial gate individually improve the performance over MVDet. However, using the channel gate and spatial gate subsequently, in other words HAM, improves in MODA and recall while retaining similar precision compared to MVDet, leading to an overall improvement in performance.

Table 6. Performance of attention modules on MVDet

Method	Wildtrack			
	MODA	MODP	precision	recall
MVDet	88.2	75.7	94.7	93.6
MVDet + Channel Gate	88.8	76.0	95.1	93.6
MVDet + Spatial Gate	88.6	**76.6**	**95.5**	93.0
MVDet + HAM	**89.4**	75.7	95.2	**94.1**

Transferability of model trained on synthetic data to real-world scenario Since real-world data can be sparse, in practice a model is often trained on synthetic data and then applied in the real world. For this reason, we compared the performance of models with and without HAM when trained on the synthetic MultiviewX data and inferenced on the real-world Wildtrack data. This experiment aims to investigate if applying HAM has any effect on performance in cross-dataset inference.

The results are shown in Table 7. Although there is noticeable improvement when HAM is applied to MVDeTr, we note that the application of HAM to SHOT does not yield any improvement for cross-dataset inference. Due to the dropping of some channels via the channel selection module, we conjecture that the domain gap between MultiviewX and Wildtrack is the cause of the performance decrease for models with multiple homographies.

Table 7. Performance evaluation on Wildtrack for MultiviewX-trained models

Method	MODA	MODP	precision	recall
MVDet	16.4	67.4	80.0	21.8
MVDet + HAM	**30.9**	66.3	**89.5**	**35.0**
SHOT	53.6	72.0	75.2	79.8
SHOT + HAM	51.6	68.0	**78.6**	70.8
BoosterSHOT	56.2	63.8	83.5	70.1

Performance under random failure of cameras In order to simulate a production environment, we test the performance of the model when cameras can fail spontaneously. We take a pre-trained instance of BoosterSHOT that is trained with all cameras and run inference on the test set of Wildtrack. During inference, we add a pre-processing step for each frame to simulate camera failure in production. First, for each frame a random number of cameras are selected to be “turned off”. The number of the “turned off” cameras is between 1 and the maximum number of available cameras (7 for Wildtrack). Then a failure rate decides if the selected cameras are failing or not. For example, for a failure rate of 0.6, anywhere between one to seven cameras fail for 60% of all frames. For failing cameras, the input images are blacked out to emulate an empty frame. We keep the decision threshold for bird’s-eye-view pedestrian detections the same because it might not be possible to adjust for camera failure in real-time. The results are shown in Table 8 and the effect of camera failure is relatively small. Even if at least one camera fails for all frames (failure rate of 1), the performance decrease is 1.2% MODA and 2.1% recall. We would normally expect camera failure to be fairly rare in production so we ran an additional experiment with a failure rate of 2%. This results in a negligible change in performance, which indicates that the method will perform consistently in a production setting. Please note that an augmentation strategy of incorporating random failing cameras into the training process is likely to increase the robustness to failing cameras. We leave such an investigation open for future work.

Table 8. Performance of BoosterSHOT on Wildtrack for random camera failure

Failure rate	MODA	MODP	precision	recall
0	91.5	82.2	96.8	94.6
0.02	91.5	82.1	97.0	94.4
0.2	90.5	82.1	96.8	93.6
0.4	90.1	82.1	96.7	93.3
0.6	89.	81.9	97.3	91.8
0.8	89.8	81.9	97.2	92.4
1	89.3	82.3	97.9	91.3

Wildtrack attention maps We show the Wildtrack attention maps for the channel selection and spatial gate modules in Figure 4. The attention maps from the channel selection module (top) resemble those from MultiviewX shown in Figure 4 in the main paper. The attention maps from the spatial selection module show different distributions from one another, which is in agreement with our claim in the main paper that different pixels in the feature map can differ in importance for each homography.

References

[1] Liangchen Song, Jialian Wu, Ming Yang, Qian Zhang, Yuan Li, and Junsong Yuan. Stacked homography transformations for multi-view pedestrian detection. In *International Conference on Computer Vision (ICCV)*, 2021. 1, 3

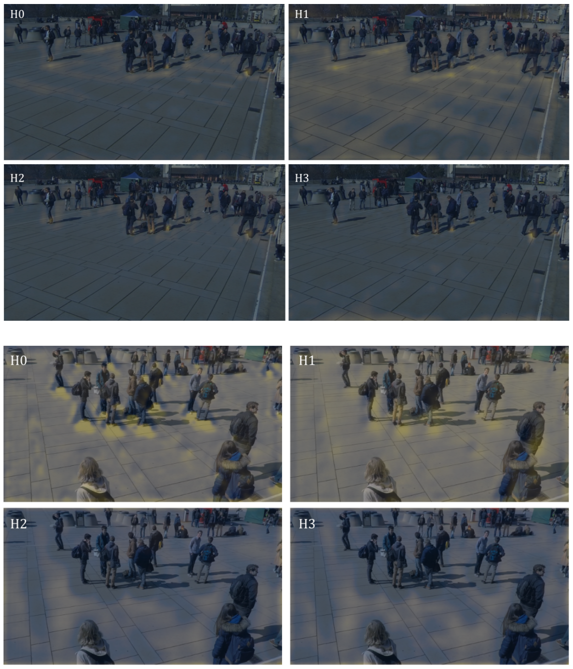

Figure 4. Homography-wise output from channel selection (top) and spatial attention maps (bottom)