Calibration-free Multi-view Crowd Counting Supplemental

Qi Zhang[1,2] and Antoni B. Chan[2]

[1] College of Computer Science & Software Engineering, Shenzhen University, China
[2] Department of Computer Science, City University of Hong Kong, Hong Kong SAR, China
{qzhang364-c@my., abchan@}cityu.edu.hk

1 Extra experiments

We train the proposed models with ground-truth homography and ground-truth density map, whose results are shown in Table 1. It shows the performance can be increased a lot with ground-truth density map, which indicates the good potential of the proposed framework if better single-view density map prediction can be obtained. And the proposed CF-MVCC-C can also achieve better performance than CF-MVCC.

Besides, it is not suitable to directly use SVC methods for the multi-view counting task, which targets at large scenes that cannot be covered by one camera. To show this, we use the **ground-truth** single-view density maps as the "oracle" single-view counting method with various baselines, and the results on CVCS dataset are in Table 3. The first two baselines totally neglect the geometry between camera views, and thus are not suitable for multi-view crowd counting. Our method (with predicted density maps) performs better than Dmap_weightedH w/ oracle SVC because our method performs matching between views to handle occlusions.

We directly test the proposed model trained on the large synthetic dataset on the CityStreet dataset with ground-truth and predicted homography matrix, shown in Table 3. The testing performance is improved using ground-truth homography matrix, but not significant enough, which also shows the homography estimation is not the bottleneck of the adaptation performance to real scenes. Therefore, we finetune the single-view counting network for better performance.

2 Visualization results

We show the example of the projection in Fig. 1, and the predicted confidence maps C, weight maps W, density maps D and matching scores M_{ij} in Fig. 2.

Projection. The example of the projection with ground-truth homography matrix and with predicted homography matrix is shown in Fig. 1. Each column shows the projected images from other camera views to View $i (i = [1, 5])$. From the example, we can observe that the projection with the predicted homogrphy

Setting	Method	MAE	NAE
H_{gt}, D_{pred}	CF-MVCC	12.04	0.101
	CF-MVCC-C	11.69	0.098
H_{gt}, D_{gt}	CF-MVCC	2.90	0.025
	CF-MVCC-C	2.79	0.023

Table 1. Ablation study on the density map input. D_{gt} means replace the predicted density map D_{pred} with the ground-truth map in Eq. 5.

Method	MAE	NAE
randomly use one view's prediction (oracle)	54.87	0.465
average of all views' predictions (oracle)	53.72	0.455
Dmap_weightedH (oracle)	16.70	0.143
Ours with CSRnet backbone (predicted)	13.90	0.118
Ours with LCC backbone (predicted)	12.79	0.109

Table 2. Comparison between SVC baselines using oracle (GT) density maps and our method using predicted density maps.

matrix can generally align the same people in different camera views, which shows the effectiveness of the homography estimation module.

Confidence, weight maps and matching scores. It can be observed that, for regions in the box, which cannot be seen by other cameras, so their weights are large regardless of the confidence scores; For the person in the red circles, which can be seen the the 3 camera views (3, 4 and 5), the weight is small due to being seen by multiple cameras.

Even though the matching scores contain some errors, but it's reasonable because no pixel-level supervision is used in the view-pair matching CNNs. Besides, only the density map region's matching score is effective in final count calculation and the background or other non-people objects' matching results are not used.

3 Failure case analysis

We show an example of the failure cases of the homography estimation module in Fig. 3. The possible reason might be the people are too far away in the camera view pairs, which is difficult for the homography estimation model to match them. Besides, there are many similar objects (eg. trees) in the scene, which is confusing to distinguish them and find correspondences between camera views.

4 Layer settings

We show the layer settings for the model in Table 4, 5, 6, and 7, including single-view counting (SVC), homography estimation CNNs and view-pair matching CNNs (VPM), and the confidence map estimation CNNs (WMP).

H_{gt}		H_{pred}	
MAE	NAE	MAE	NAE
40.76	0.501	48.58	0.602

Table 3. The performance of directly testing the proposed model trained on the large synthetic dataset on the real dataset CityStreet with ground-truth and predicted homography matrix. The table shows the homography estimation is not the bottleneck of the adaptation performance to real scenes.

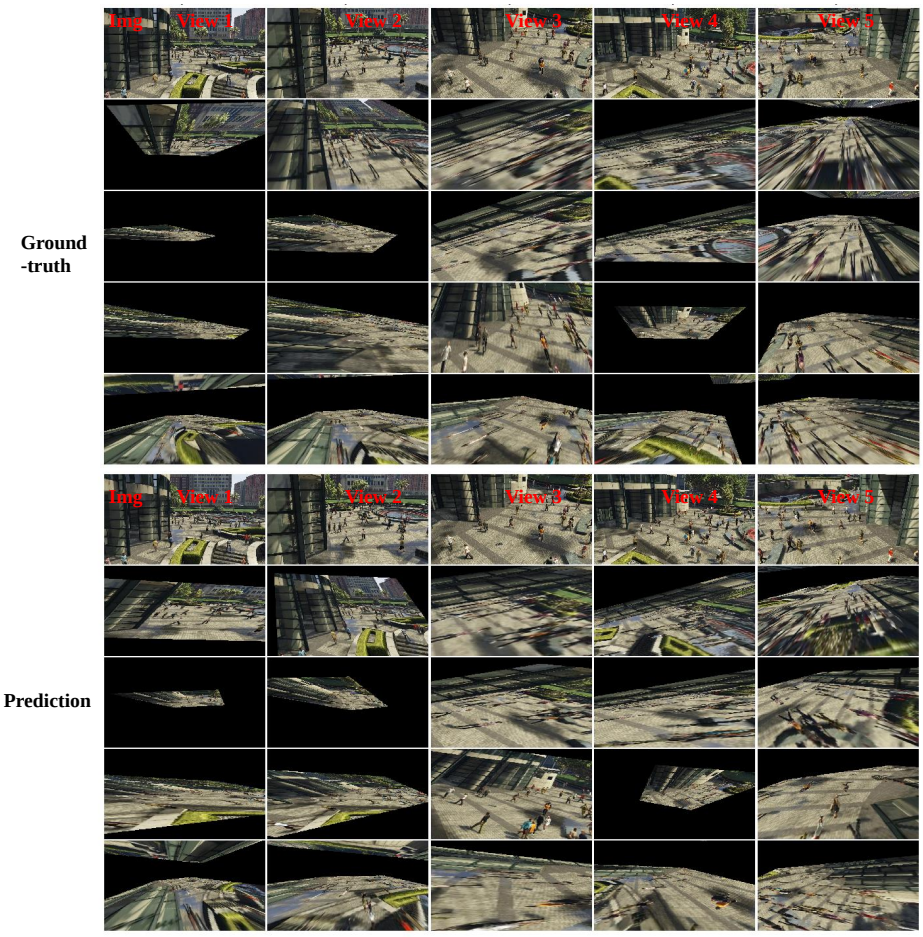

Fig. 1. Example of the ground-truth and prediction projection. Each column shows the projected images from other camera views to View $i(i = [1, 5])$.

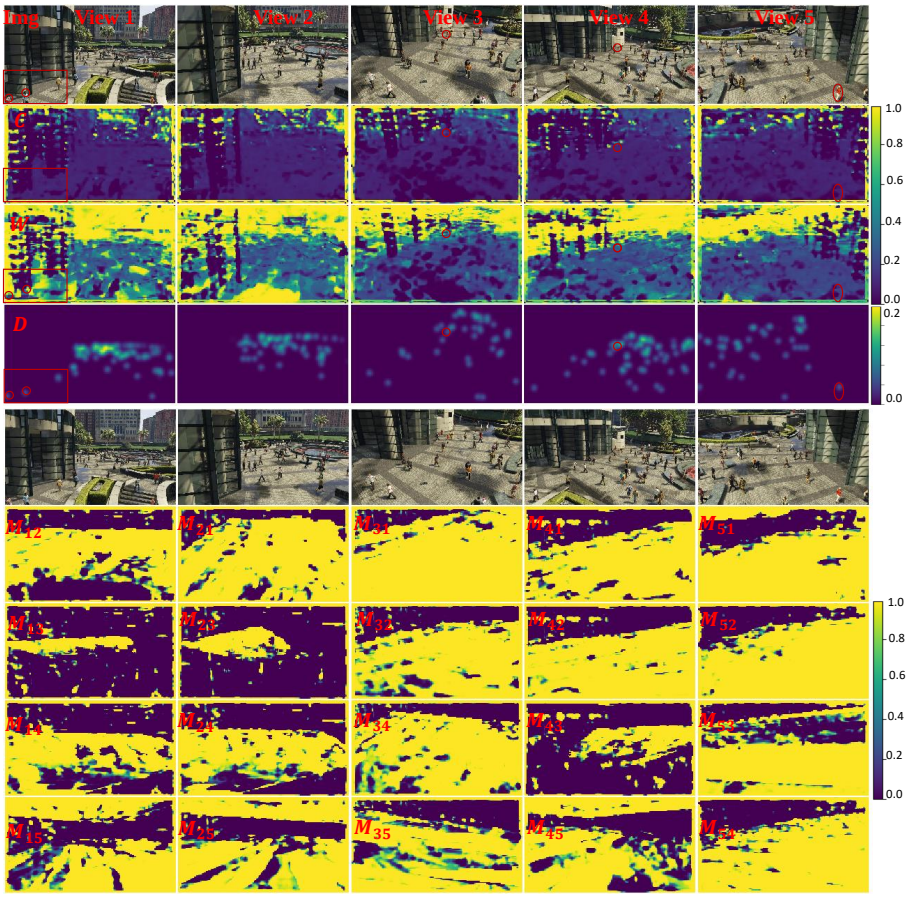

Fig. 2. Example of the predicted confidence maps C, weight maps W, density maps D and matching result M_{ij}. It can be observed that, for regions in the box, which cannot be seen by other cameras, so their weights are large regardless of the confidence scores.

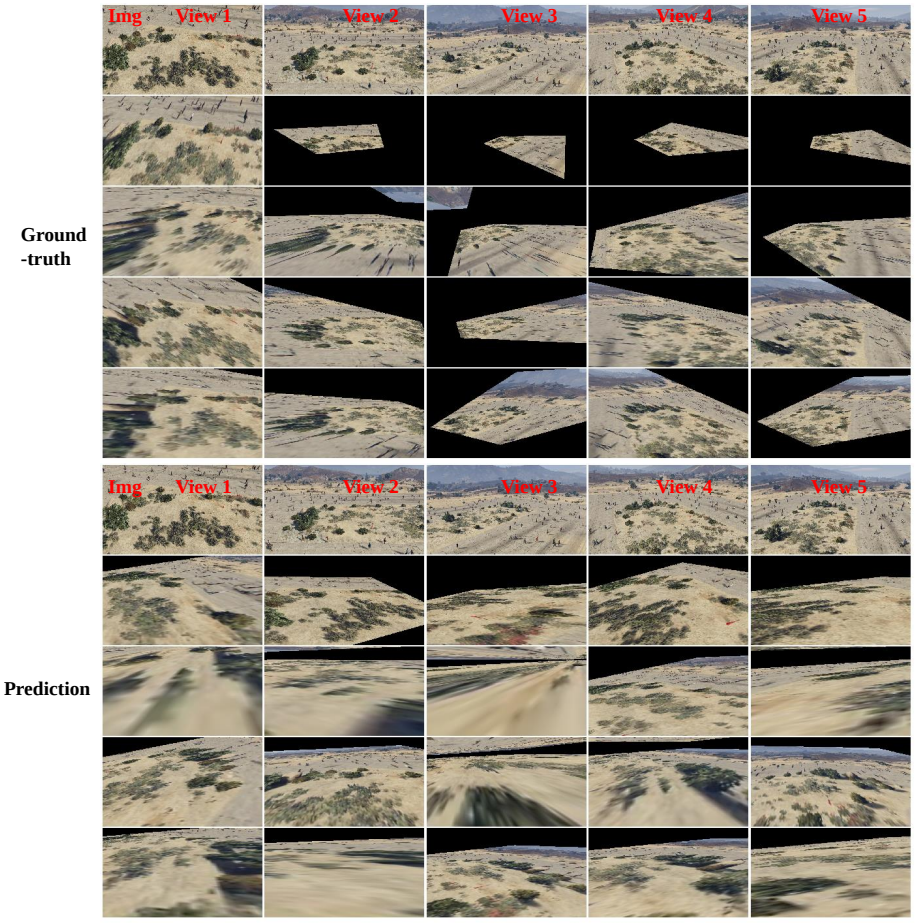

Fig. 3. Example of the failure cases of the homography estimation module.

Counting feature extractor F^c	
Layer	Filter
conv 1	$64\times3\times3\times3$, relu
conv 2	$64\times64\times3\times3$, relu
pooling	2×2
conv 3	$128\times64\times3\times3$, relu
conv 4	$128\times128\times3\times3$, relu
pooling	2×2
conv 5	$256\times256\times3\times3$, relu
conv 6	$256\times256\times3\times3$, relu
conv 7	$256\times256\times3\times3$, relu
Counting decoder	
Layer	Filter
conv 1	$512\times256\times3\times3$, relu
conv 2	$512\times512\times3\times3$, relu
conv 3	$512\times512\times3\times3$, relu
conv 4	$512\times512\times3\times3$, relu, d=2
conv 5	$512\times512\times3\times3$, relu, d=2
conv 6	$512\times512\times3\times3$, relu, d=2
conv 7	$256\times512\times3\times3$, relu, d=2
conv 8	$128\times256\times3\times3$, relu, d=2
conv 9	$64\times128\times3\times3$, relu, d=2
conv 10	$1\times64\times1\times1$, relu

Table 4. The feature extractor and decoder of single-view counting module (SVC). The Filter dimensions are output channels, input channels, and filter size ($w\times h$). 'd' means the dilation rate, and if not specified, it's 1.

Feature extractor F^h	
Layer	Filter
conv 1	$64\times3\times3\times3$, relu
conv 2	$64\times64\times3\times3$, relu
pooling	2×2
conv 3	$128\times64\times3\times3$, relu
conv 4	$128\times128\times3\times3$, relu
pooling	2×2
conv 5	$256\times256\times3\times3$, relu
conv 6	$256\times256\times3\times3$, relu
conv 7	$256\times256\times3\times3$, relu
pooling	2×2
Decoder	
Layer	Filter
correlation	-
conv 1	$64\times n\times1\times1$, relu
conv 2	$64\times64\times3\times3$, relu
conv 3	$32\times64\times3\times3$, relu
conv 4	$1\times32\times1\times1$, relu
flatten	-
fc 1	64
fc 2	8

Table 5. The feature extractor and decoder of homography estimation CNNs. 'n' is the output channel size of the correlation layer, decided by the input feature map size. 'fc' means the fully-connected layer, and the parameter initialization of 'fc 2' is $w = 0, b = [1, 0, 0, 0, 1, 0, 0, 0]$

View-pair matching CNNs	
Layer	Filter
projection layer	spatial transformation layer
concatenation	-
conv 1	$128\times512\times3\times3$, relu
conv 2	$64\times128\times3\times3$, relu
conv 3	$1\times64\times1\times1$, relu

Table 6. The view-pair matching CNNs. The concatenation layer's output channel size is 512. The Filter dimensions are output channels, input channels, and filter size ($w\times h$).

Distance feature extractor T	
Layer	Filter
conv 1	$128\times1\times3\times3$, relu
conv 2	$64\times128\times3\times3$, relu
Confidence decoder	
Layer	Filter
concatenation	-
conv 1	$128\times320\times3\times3$, relu
conv 2	$64\times128\times3\times3$, relu
conv 3	$1\times32\times1\times1$, relu

Table 7. The distance feature extractor and confidence decoder. The concatenation layer's output channel size is 320 (256+64).