3D Crowd Counting via Multi-View Fusion with 3D Gaussian Kernels

Qi Zhang, Antoni B. Chan
Department of Computer Science, City University of Hong Kong
Tat Chee 83, Kowloon Tong, Hong Kong SAR, China
qzhang364-c@my.cityu.edu.hk, abchan@cityu.edu.hk

Abstract

Crowd counting has been studied for decades and a lot of works have achieved good performance, especially the DNNs-based density map estimation methods. Most existing crowd counting works focus on single-view counting, while few works have studied multi-view counting for large and wide scenes, where multiple cameras are used. Recently, an end-to-end multi-view crowd counting method called multi-view multi-scale (MVMS) has been proposed, which fuses multiple camera views using a CNN to predict a 2D scene-level density map on the ground-plane. Unlike MVMS, we propose to solve the multi-view crowd counting task through 3D feature fusion with 3D scene-level density maps, instead of the 2D ground-plane ones. Compared to 2D fusion, the 3D fusion extracts more information of the people along z-dimension (height), which helps to solve the scale variations across multiple views. The 3D density maps still preserve the 2D density maps property that the sum is the count, while also providing 3D information about the crowd density. We also explore the projection consistency among the 3D prediction and the ground-truth in the 2D views to further enhance the counting performance. The proposed method is tested on 3 multi-view counting datasets and achieves better or comparable counting performance to the state-of-the-art.

1 Introduction

Single-view crowd counting has been studied for decades and has achieved promising performance on the existing counting datasets (**?**; **?**; **?**; **?**; **?**; **?**). However, in real-world applications, there are several situations where single-view counting cannot perform well (e.g., see Fig. 1): 1) The scene is too wide (such as a park or a football match stadium) where a single camera's field-of-view is limited and cannot cover the whole scene; 2) The scene is too long (such as the underground train platform) where the people who are far away from the camera have very low resolution and the counting methods' performance drops on these people; 3) The scene contains a lot of obstacles, such as vehicles, building structures or *etc.*, where many people are heavily or totally occluded. Under the 3 conditions, the current single-view counting methods are inaccurate, because many people are miscounted due to limited field-of-view, low resolution and severe occlusion.

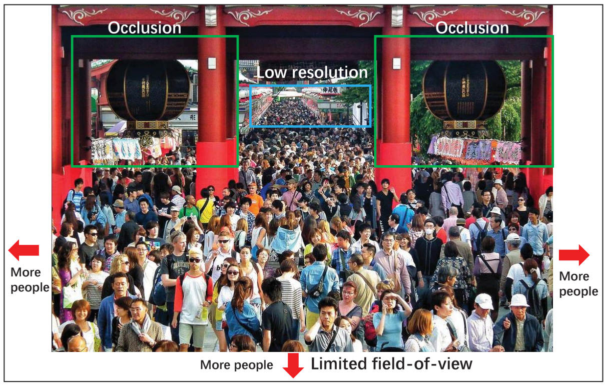

Figure 1: An example for the limitation of single-view counting for large and wide scenes: limited field-of-view, low resolution and severe occlusion. This image is from the ShanghaiTech dataset (**?**).

To address the aforementioned situations, multiple cameras should be deployed, and the multi-view information should be fused to enhance the counting performance for complicated scenes. A few works (**?**; **?**; **?**; **?**) have considered multi-view counting. However, the hand-crafted features and the foreground extraction steps limit their performance. Recently, a CNN-based multi-view counting method (**?**) was proposed, which improved the counting performance on wide-area scenes. In their paper, the multi-view information (camera-view density maps or feature maps) are projected to the same 2D plane in the 3D scene (at the height of a person), and then fused to predict the 2D scene-level density maps on the ground-plane. Several fusion methods are considered, including late and early fusion, and a multi-view multi-scale early fusion model (MVMS), which handles both the inter- and intra-view scale variations.

The disadvantage of the 2D-to-2D projection in (**?**) is that the features of the same person from different views may not line up correctly due to the approximation that all features come from the same height in the 3D world. Clearly this is

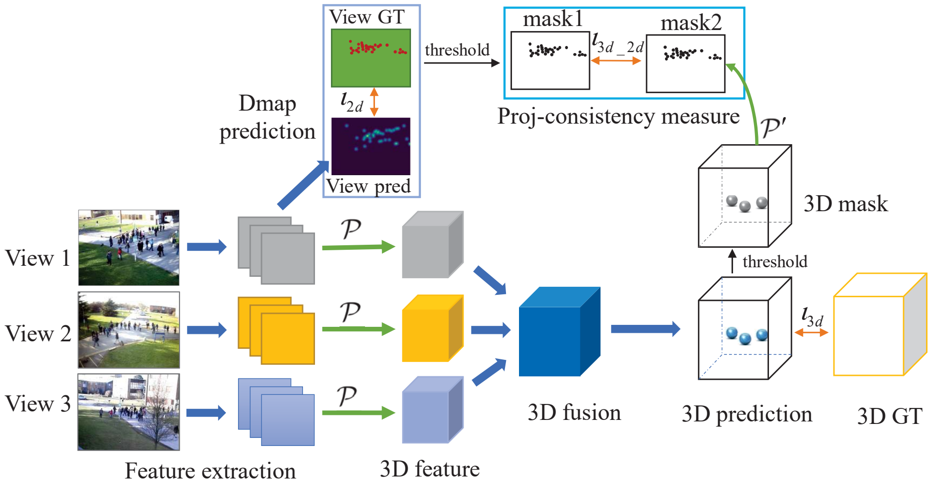

Figure 2: The pipeline of 3D crowd counting. Single-view features are extracted and then projected to the 3D world on multiple height planes. The projected 3D features are concatenated and fused to output the 3D density map prediction (loss l_{3d}). Each camera-view prediction branch decode the 2D features to obtain the 2D camera-view predictions (loss l_{2d}). Finally, the 3D prediction is back-projected to each camera-view, and the the projection consistency between the camera-view ground-truth and the back-projected prediction is measured (loss l_{3d_2d}).

not true for features extracted from the heads and feet of the people. To address this problem, in this paper, we propose to use 3D projection and 3D feature fusion to perform the multi-view counting task. Our proposed method consists of following components (see Fig. 2): 1) *Single view feature extraction and density map prediction*: 2D single-view features are extracted and then decoded to the 2D density map predictions; 2) *2D-to-3D projection*: A differentiable projection layer together with the camera parameters are used to perform the fixed 3D projection from image plane to 3D world; 3) *3D fusion and prediction*: The projected multi-view 3D features are fused using 3D CNN layers to predict the 3D scene-level density map ; 4) *Projection consistency between the 3D prediction and 2D views*: The 3D prediction is back-projected to each camera view and a loss between the camera view prediction and back-projected 3D prediction is added to enhance the multi-view counting performance.

In our approach, the 3D projection provides more information of the people, and the projected people's scales along the z-dimension in the 3D world are similar to each other (decided by the people height), which suggests the scale variation issue among multiple views can be tackled without the scale-selection module like in MVMS. In addition, a 3D density map with 3D Gaussian kernels is used to represent the crowd in the scene, instead of the 2D scene-level density maps. The 3D density map can provide more information about the crowd in 3D, e.g., the elevation of the crowd.

In summary, the main contributions of our paper are:

- We propose an end-to-end DNNs-based 3D multi-view counting method that predicts 3D density maps, which provides information about the crowd in 3D. To the best of our knowledge, this is the first study of 3D scene-level multi-view crowd counting.
- Unlike the previous methods, we use 3D projection and 3D fusion, which can help to deal with the scale variation issue without the scale selection module.
- The projection consistency between the camera-view ground-truth and the back-projected 3D predictions is explored to further boost the multi-view counting performance. The proposed method can achieve better or comparable counting performance to the state-of-art.

2 Related Work

In this section, we will review the existing single-view and multi-view counting methods. We also review 3D object reconstruction using deep neural networks (DNNs).

2.1 Single-view counting

Single-view crowd counting has achieved satisfactory performance on the existing counting datasets, especially the DNN-based density map estimation methods. **? ?** proposed a single-column CNN framework to directly estimate the density maps from the images. Scale variations due to perspective changes is a critical issue in the crowd counting task, which can limit the performance, and many methods have been proposed to handle scale variations (**?**; **?**; **?**; **?**; **?**). **? ?** proposed the contextual pyramid CNN (CP-CNN) to incorporate global and local context information in the crowd counting framework. Furthermore, extra information and more sophisticated networks were utilized to further improve the counting performance (**?**; **?**; **?**; **?**; **?**; **?**; **?**; **?**; **?**). **? ?** proposed an adaptive convolution neural network (ACNN) by utilizing the context (camera height and angle) as side information in the counting framework. **? ?** integrated the perspective information to provide additional knowledge of the person scale change in an image. **? ?** proposed a regression guided detection network (RDNet) for RGB-D crowd counting. **? ?** proposed Recurrent Attentive

Zooming Network to zoom high density regions for higher-precision counting and localization.

The single-view based counting methods cannot handle the situations when the camera cannot cover the whole scene, the occlusions in the scene are too severe, or the people are in low resolution due to long distance to the camera. Therefore, multiple cameras should be adopted to improve the counting performance for these wide-area scenes.

2.2 Multi-view counting

Traditional multi-view counting methods can be divided into 3 categories: detection/tracking based (**?**; **?**; **?**; **?**), regression based (**?**; **?**), 3D cylinder based (**?**). These multi-view counting methods have several limitations. First, they need to utilize foreground extraction techniques to segment the crowd from background. Therefore, the effectiveness of the foreground extraction step limits the final counting performance. Second, hand-crafted features are used both in the people detection or crowd count regression. The hand-crafted features lack representation ability, which reduces the robustness and the performance of the methods. Third, these methods are mainly performed and tested on the PETS2009 (**?**), which is a multi-view dataset with small crowd numbers and staged crowd behavior.

Recently, **? ?** proposed a DNN-based multi-view counting method MVMS and a new larger multi-view counting dataset CityStreet. MVMS first extracts camera-view information (density maps or features), and then projects them to the average-height plane in the 3D scene with the given camera parameters. Next, the projected features are fused and decoded to predict the scene-level density maps (on the average height plane). Our proposed method differs from MVMS in the several aspects: 1) instead of using average-height projection, we use multi-height projection, which spatially aligns the person's features (e.g., head, body and feet features) in 3D, making it easier to find the geometric correspondence across views (See Fig. 3); 2) we predict a 3D crowd density map, generated using 3D Gaussian kernels, which provides distribution of the crowd in 3D space; 3) the 3D density map prediction is back-projected to each camera view, and compared with the 2D ground-truth density map of the camera view, which defines a projection consistency loss for improving the accuracy.

2.3 DNN-based 3D reconstruction

Our 3D crowd counting method is related to a few works on DNN-based 3D object reconstruction and human pose estimation. **? ?** proposed an unsupervised single-view 3D object reconstruction method utilizing the projection loss defined by the perspective transformation. **? ?** proposed a 3D RNNs architecture to gradually refine the multi-view 3D object reconstruction step-by-step. **? ?** leveraged the 3D geometry constraint of features through projection and unprojection operations in the 3D reconstruction method. **? ?** presented DeepMVS to produce a set of plane-sweep volumes and then predict high-quality disparity maps. **? ?** proposed two learnable triangulation methods for 3D human pose estimation: algebraic triangulation and volumetric aggregation.

It can be observed that the existing DNNs-based 3D reconstruction methods are mainly focused on the reconstruction of a single object (see datasets ShapeNet (**?**) and IKEA (**?**), or human pose). Our proposed method can also be regarded as predicting a 3D representation from multiple viewpoints, but differs from the existing DNN-based 3D object reconstruction in several aspects. First, the proposed method can do more than 3D shape reconstruction, because the 3D Gaussian kernels can represent the crowd's 3D distribution as well as indicate the crowd count. Second, unlike the previous 3D object reconstruction, the proposed method targets at the scene-level representation (all people in the scene), not only a single object. Furthermore, we exploit the relationship between the 2D camera-view density maps and 3D scene-level density maps to obtain a projection consistency loss.

3 3D Counting via 3D Projection and Fusion

We follow the setup of multi-view counting in **? ?**: we assume fixed cameras with known intrinsic and extrinsic camera parameters, and synchronized camera frames across views. In contrast to **? ?**, which is based on 2D ground-plane density maps, we generate 3D ground-truth density maps by convolving the 3D ground-truth annotations with fixed-width 3D Gaussian kernels. The 3D ground-truth annotation coordinates are calculated from the 2D view annotations and people correspondence across views (see Section 4 for more details).

In this section, we introduce the end-to-end DNNs-based 3D crowd counting method consisting of following stages. 1) *Single view feature extraction and density map prediction*: 2D single-view features are extracted and decoded to predict a 2D camera-view density map; 2) *2D-to-3D projection*: a differentiable projection layer using the camera parameters projects the camera-view feature maps from the image plane to multiple height-planes in the 3D world; 3) *3D fusion and prediction*: the projected multi-view 3D features are fused to predict the 3D density maps using 3D CNN layers; 4) *Projection consistency between the 3D prediction and 2D views*: the 3D prediction is back-projected to each camera view, and then compared with the corresponding camera-view 2D ground-truth using loss to refine the 3D prediction.

3.1 Single-view branch

Each camera-view image is fed into a CNN to extract the camera-view features. The 2D camera-view features are decoded by another CNN to predict the camera-view density maps. The camera-view feature extraction and density map prediction layer settings can be found in Table 1 left.

Here, the camera-view prediction branches contribute to the model performance in 2 aspects. First, the camera-view prediction branches together with the 2D density map supervision improves the training of the camera-view feature extraction. A similar strategy is used in **? ?**, where the camera-view prediction branches are used in the first stage for pretraining. Second, the camera-view prediction branches force the 2D feature representation to be consistent with the 3D feature representation, and the difference between them is

Single-view branch	
Layer	Filter
Feature extraction	
conv 1	$16\times 1\times 5\times 5$
conv 2	$16\times 16\times 5\times 5$
pooling	2×2
conv 3	$32\times 16\times 5\times 5$
conv 4	$32\times 32\times 5\times 5$
pooling	2×2
Density map prediction	
conv 5	$64\times 32\times 5\times 5$
conv 6	$32\times 64\times 5\times 5$
conv 7	$1\times 32\times 5\times 5$

3D Fusion module	
Layer	Filter
concatenation	-
3D conv 1	$32\times n\times 5\times 5\times 7$
3D conv 2	$64\times 32\times 5\times 5\times 7$
3D conv 3	$128\times 64\times 5\times 5\times 7$
3D conv 4	$64\times 128\times 5\times 5\times 7$
3D conv 5	$32\times 64\times 5\times 5\times 7$
3D conv 6	$32\times 32\times 5\times 5\times 7$
3D conv 7	$1\times 32\times 5\times 5\times 7$

Table 1: The layer settings for the camera-view feature extraction and density map prediction branches (left), and the 3D fusion module (right). The filter dimensions are output channels, input channels, and filter size $w_0\times h_0\times d_0$ ($d_0 = 1$ in 2D conv layers).

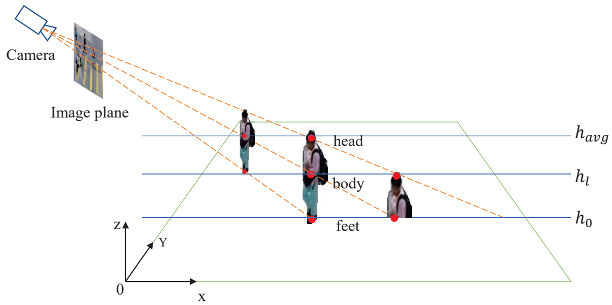

Figure 3: The multi-height projection can extract feature of the people body along z dimension and form a 3D feature representation for the people, which is consistent with the 3D scene.

the geometric projection. This constraint is natural because the 2D and 3D observations are linked by the geometry projection, and the same link still exists between the 2D and 3D feature representations. The 2D-3D feature representation constraint was also used in **? ?**, where the 3D representation is forced to be able to predict the 2D observations from it. Furthermore, this constraint can improve the final 3D prediction performance (see ablation study for more details).

Suppose there are N camera-views, the density map prediction for the i_{th} view is $V_i \in R^{h\times w}$ and the corresponding ground-truth is $V_i^t \in R^{h\times w}$, then the camera-view density map prediction loss l_{2d} is the mean-square error (MSE):

$$l_{2d} = \frac{1}{wh}\sum_{i=1}^{N} \parallel V_i - V_i^t \parallel_2^2 . \tag{1}$$

3.2 2D-to-3D projection

The extracted camera-view features are projected to the 3D world with a fixed 2D-to-3D projection layer. We assume that the camera intrinsic and extrinsic parameters are both known, which are used to perform the projection. Since each image pixel's corresponding height in the 3D world is unknown, a height range H is used in the projection, where each pixel is projected to the 3D world multiple times onto different height planes. Then, the projected features from all height are concatenated along the z-dimension to form a 3D feature representation. The 2D-to-3D projection layer can be implemented using the Sampler from the Spatial Transformer Networks (**?**) with the known camera calibration parameters (intrinsic and extrinsic). Suppose that $\mathcal{P}$ is the 2D-to-3D projection, F_i stands for the 2D feature presentation of view i, and the height range $H = \{h_0, h_1, ..., h_r\}$, then the projected 3D feature representation $(F_{3d})_i$ for view i is

$$(F_{3d})_i = \mathcal{P}_i(F_i, H) = [\mathcal{P}_i(F_i, h_0), ..., \mathcal{P}_i(F_i, h_r)], \tag{2}$$

where $[\cdot]$ is concatenation along the z-dimension (height).

In **? ?**, the fixed-height projection was used where all pixels were projected to the average height (1750mm). Compared to average-height projection, the multi-height projection can output a body feature representation along z-dim (e.g., head, body and feet features, see Fig. 3), which allows the network to better identify the location of the person through the 3D alignment of the features.

3.3 3D density map prediction

After the 2D-to-3D projection, the projected 3D features from multiple views are concatenated (along the feature channel) and then decoded by several 3D convolution layers. The architecture for the 3D feature decoder layers can be found in Table 1 right. Suppose the 3D prediction is $G \in R^{a\times b\times d}$ and the corresponding ground-truth is $G^t \in R^{a\times b\times d}$, the 3D prediction loss l_{3d} is the MSE

$$l_{3d} = \frac{1}{abd} \parallel G - G^t \parallel_2^2 . \tag{3}$$

3.4 3D-to-2D projection consistency measure

Considering the geometry constraint between 2D observations and the 3D observation, we also require the 3D prediction be consistent with the 2D single view density maps in terms of projection geometry. To achieve this, the 3D prediction G is projected to each camera-view i with a 3D-to-2D projection operation $\mathcal{P}'$. The projection consistency between the projected 2D density maps and the 2D density map ground-truth is measured and used as part of the loss to further enhance the 3D counting performance.

3D-to-2D projection. Suppose the $\mathcal{P}'$ is the 3D-to-2D projection, G_{mask} is a 3D binary mask which is the 3D prediction G thresholded at $T = 10^{-4}$. When performing the projection, each pixel's height of the 2D view plane is also unknown (each 2D image point is corresponding to many 3D points along the view ray without the height information). Therefore, the height range $H = \{h_0, h_1, ..., h_r\}$ is also used in the 3D-to-2D projection. Then the projected 2D mask $(G_{2dmask})_i$ for view i can be denoted as,

$$(G_{2dmask})_i = \text{sign}(\sum_{l=0}^{r} \mathcal{P}_i'(G_{mask}, h_l)), \tag{4}$$

where $\sum$ is summation along the z-dimension (height dimension). In the 3D-to-2D projection, the projected 2D pixel

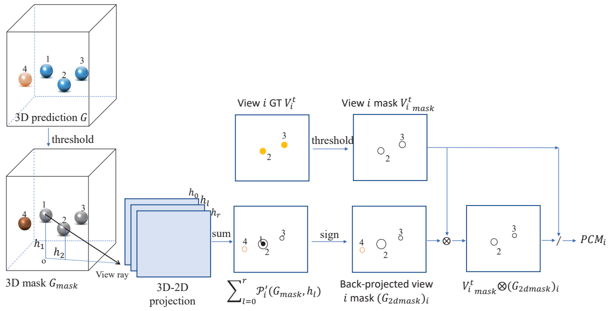
3D prediction G
threshold
3D mask G_{mask}
View ray
h_1
h_2
o
h_0
h_l
h_r
3D-2D projection
sum
$\sum_{l=0}^{r} \mathcal{P}'_i(G_{mask}, h_l)$
View i GT V_i^t
threshold
View i mask $V^t_{i\ mask}$
sign
Back-projected view i mask $(G_{2dmask})_i$
$V^t_{i\ mask} \otimes (G_{2dmask})_i$
PCM_i
1
2
3
4

Figure 4: Example of projection consistency measure (PCM). There are 4 people in the 3D prediction, while only Person 2 and Person 3 are visible in the 2D view i. Since Person 1 is occluded by Person 2 and Person 4 is totally occluded in view i, they are masked out in the PCM calculation. When the people location in the 3D prediction is not consistent with the 2D ground-truth, the PCM value is low.

value can be regarded as the binary summation along the view ray in the 3D prediction grid (see Fig. 4).

Projection consistency measure. The main challenge to measure the consistency between the 3D density map prediction and the 2D camera-view density map is that some people who are occluded in a 2D camera-view will be present in the 3D prediction (because they are visible from other views). Thus the MSE loss cannot be used directly here. Instead, we measure the consistency between the binary masks produced by thresholding the 3D prediction and the corresponding 2D ground-truth density map, while accounting for inconsistencies due to occlusions (see Fig. 4).

We define the 3D-to-2D projection consistency measure (PCM) for view i:

$$PCM_i = \frac{\parallel V^t_{i\ mask} \otimes (G_{2dmask})_i \parallel_1}{\parallel V^t_{i\ mask} \parallel_1 + \alpha}, \tag{5}$$

where $\otimes$ denotes element-wise multiplication, $V^t_{i\ mask}$ is a binary mask which is the ground-truth density map of view i thresholded at 1e-3, $(G_{2dmask})_i$ is the back-projected 3D prediction mask, and α=1e-5 is a constant to prevent divide-by-zero. The important property of the PCM is that no penalty occurs when an extra person is present in the 3D prediction ($(G_{2dmask})_i$), but not the 2D camera-view (e.g., due to occlusion). On the other hand, the PCM will be reduced when a person in the 2D camera-view is missing in the 3D prediction. Finally, the projection consistency loss is defined by summing over the camera-views:

$$l_{3d\_2d} = \sum_{i=1}^{N} (1 - PCM_i). \tag{6}$$

3.5 Training loss

Combining all the aforementioned losses, the final loss is

$$l_{all} = l_{3d} + \beta l_{2d} + \gamma l_{3d\_2d}, \tag{7}$$

where β and γ are hyperparameters for weighting the contributions of each term.

4 Experiments and Evaluation

In this section, we will discuss the implementation details, and conduct experiments of our proposed 3D crowd counting framework on three multi-view counting datasets.

4.1 Implementation details

3D ground-truth generation. The 3D ground-truth is generated with the single-view annotations (here we use head annotations) and the cross-view correspondence. Suppose the same person's corresponding annotations in M views ($M \leq N$) are $\{(x_j, y_j)\}, j \in \{1, 2, ..., M\}$, their corresponding 3D coordinates are denoted as $\{W_j\} = \{\mathcal{P}_j((x_j, y_j, h))\}$ where the height h is not known. Considering the projection correspondence, the 3D coordinate of the person's head $W = [X, Y, Z]^T$ can be obtained by finding the height h that minimizes the spread of the projected head positions from all views,

$$Z = \underset{h}{\text{argmin}} \parallel W_j - \overline{W} \parallel_2^2, \tag{8}$$

$$(X, Y) = \frac{1}{M} \sum_{j=1}^{M} \mathcal{P}_j((x_j, y_j, Z)), \tag{9}$$

where $\overline{W}$ is the mean of $\{W_j\}$. Z is the height h minimizing (8), which can be obtained through searching h in the height range $H_0 \in \{h_0, h_1, ..., h_{r0}\}$ (we use range $\{1000, 1010, 1020, ..., 2000mm\}$). For numerical stability during training, 2D density ground-truth maps are scaled by 10^3, while the 3D ground-truth is scaled by 10^4. The predictions are scaled downward correspondingly for evaluation.

Training settings and methods. In the proposed method (see Fig. 2), except the 3D counting prediction, there are two important modules: single-view density map prediction and the projection consistency measure module. In the ablation study, we show the two modules' influence to the final 3D counting performance. A 3-stage training is used for the proposed method: In the first stage, $\beta = 1$ means that the single-view 2D supervision is dominant to benefit the feature extraction training; In the second stage, $\beta = 0.01$ means the 3D supervision is dominant to accelerate the 3D fusion training; In the third stage, $\beta = 0.01$ and γ is variable according to ground-truth setting, which increases the influence of the projection consistency measure (PCM) to further enhance the performance. In all stages, the learning rate is set as 10^{-4} and the batch size is 1.

4.2 Evaluation methods

Datasets. The proposed method is evaluated and compared on the 3 multi-view counting datasets, PETS2009 (**?**), DukeMTMC (**?**) and CityStreet (**?**). We directly use the same dataset settings as in **? ?**. PETS2009 contains 3 views, and 1105 and 794 images are for training and testing, respectively. The input image resolution ($w \times h \times d$) is 384×288 and the 3D ground-truth resolution is $152 \times 177 \times 7$. The voxel height in z-dim is 40cm in the real world and the height range is 0-2.8m. DukeMTMC (**?**) contains 4 views, and the first 700 images are used for training and remaining 289 images are for testing. The input image resolution is 640×360

Dataset	PETS2009	DukeMTMC	CityStreet
Dmap weighted	8.32	2.12	9.36
Detection+ReID	9.41	2.20	27.60
Late fusion (**?**)	3.92	1.27	8.12
Naïve early fusion (**?**)	5.43	1.25	8.10
MVMS (**?**)	3.49	**1.03**	8.01
3D counting (ours)	**3.15**	1.37	**7.54**

Table 2: Experiment results: mean absolute error (MAE) on three multi-view counting datasets. “3D counting” uses both single-view branches and the projection consistency measure loss.

Dataset	PETS2009			DukeMTMC			CityStreet		
$n * h$	3D	3D+2D	3D+2D+PCM	3D	3D+2D	3D+2D+PCM	3D	3D+2D	3D+2D+PCM
7*40cm	4.12	3.20	**3.15** ($\gamma = 100$)	1.82	1.71	1.65 ($\gamma = 10$)	8.98	8.49	8.35 ($\gamma = 100$)
14*20cm	4.88	4.57	4.24 ($\gamma = 10$)	2.12	1.63	1.49 ($\gamma = 3$)	8.72	7.89	7.71 ($\gamma = 30$)
28*10cm	5.34	4.27	4.21 ($\gamma = 1$)	2.15	1.41	**1.37** ($\gamma = 0.5$)	7.87	7.58	**7.54** ($\gamma = 10$)

Table 3: Ablation study of the training loss and ground-truth settings. γ is the hyperparameter for the projection consistency measure (PCM) loss. n is the number of voxels in the z-dimension (height), and h is the voxel height in the 3D world. For DukeMTMC, the number of voxels in z-dimension is slightly larger (9, 18, 36).

and the 3D ground-truth resolution is $160 \times 120 \times 36$. The voxel height in z-dim is 10cm in the real world and the height range is 0-3.6m. CityStreet (**?**) consists of 3 views and 500 images in which the first 300 for training and rest 200 for testing. The input image resolution is 676×380 and the 3D ground-truth resolution is $160 \times 192 \times 28$. The voxel height in z-dim is 10cm in the real world and the height range is 0-2.8m.

Comparison methods. 5 multi-view counting methods are used for comparison. 1) “Dmap weighted” fuses single-view density maps into a scene-level count with a view-specific weighted map, which is constructed based on how many views can see a particular pixel. In the experiment, CSR-net (**?**) is used to predict the single-view density maps for datatsets PETS2009 and CityStreet, and FCN-7 for DukeMTMC. 2) “Detection + ReID” first detects all humans in each camera-view and then the scene geometry constraints and person ReID are used to associate the same people across views. Specifically, Faster-RCNN (**?**) is used for people detection and LOMO 2015 (**?**) for person ReID. 3) “late fusion” model fuses single-view density maps to predict 2D height-plane density maps (**?**); 4) “naïve early fusion” model fuses feature maps to predict 2D height-plane density maps (**?**); 5) “multi-view multi-scale (MVMS)” model fuses feature maps with a scale selection module to cope with the scale variation issue (**?**).

Evaluation metric. The mean absolute error (MAE) is used to evaluate the scene-level counting performance, comparing the scene-level predicted and ground-truth counts.

4.3 Experiment results

The experimental results are shown in Table 2 and the visualization results can be found in the Fig. 5 (also see the Supplemental). On **PETS2009**, the proposed 3D multi-view counting method can achieve better results than the two baseline multi-view counting methods (“Dmap weighted”, “Detection + ReID”) and the 3 versions of the end-to-end multi-view counting method proposed in **? ?**. The first two baseline methods cannot effectively fuse the multi-view information, which limits their performance. The proposed method achieves better performance than MVMS (**?**), which shows the advantage of the 3D projection and 3D fusion. On **DukeMTMC**, our 3D multi-view counting method achieves comparable performance to the MVMS (**?**). But the proposed method still achieves better performance than the two baseline methods. Due to low crowd count and lack of occlusions in the DukeMTMC, the performance gap is not very obvious. On **CityStreet**, our 3D multi-view counting method achieves the best results among the end-to-end multi-view counting method (late fusion, naive early fusion, MVMS) and the two baseline methods. “Detection + ReID” performs badly on CityStreet due to large crowd count and severe occlusions.

4.4 Ablation study

In this section, we perform ablation studies on the training loss and the ground-truth settings.

Training loss. The rows of Table 3 show the results of using 3D loss, 3D+2D loss or 3D+2D+PCM loss on the 3 datasets. It can be observed using single-view prediction branches and 2D supervision (3D+2D loss) can achieve better multi-view counting performance in comparison with only using 3D loss. Furthermore, using 3D+2D together with PCM loss can obtain better multi-view counting performance compared to using only 3D loss or 3D+2D loss on all 3 datasets with different ground-truth settings.

Ground-truth setting. The columns of Table 3 show the results of using different resolution of the 3D density map (n is the number of voxels in the z-dimension, and h is the voxel height in the 3D world). For PETS2009, the best performance is achieved by using voxel number 7 and voxel height 40cm with 3D+2D+PCM loss ($\gamma = 100$). For DukeMTMC, using voxel number 36 and voxel height 10cm with 3D+2D+PCM loss ($\gamma = 0.5$) gives the best result. As to CityStreet, the best result is obtained by using voxel number 28 and voxel height 10cm with 3D+2D+PCM loss ($\gamma = 10$).

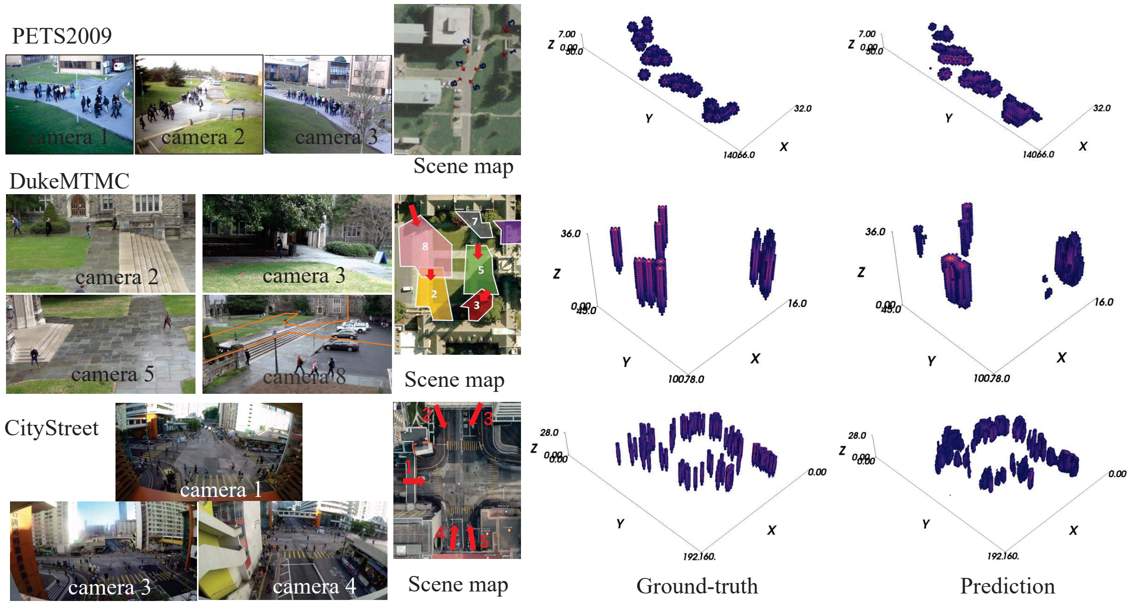

Figure 5: Examples of the 3 multi-view datasets and their prediction results. The 3D density maps of PETS2009, DukeMTMC and CityStreet are thresholded by 5e-3, 1e-3 and 1e-3, respectively. See the supplemental for more visualizations.

Compared to DukeMTMC and CityStreet, the best performance of PETS2009 is achieved at h=40cm. The people occlusion in PETS2009 is more severe and many people's lower bodies are totally occluded from all views (e.g., the people in the middle). Thus, increasing the height resolution does not provide additional information of the body, but may introduce more noises (other people's features) along the z-dim, thus leading to worse performance.

5 Conclusion and Discussion

In this paper, a DNN-based 3D multi-view counting method is proposed, which fuses camera-views to predict the 3D scene-level density map. 3D projection and fusion are used, which can handle the situation when people are not all located at the same height (e.g., people standing on a staircase), and provides a chance to solve the scale variation issue in the 3D space without a scale selection operation. The projection consistency measure between the 3D prediction and 2D density map ground-truth is studied and then utilized in the loss function to refine the 3D prediction further. Compared to other state-of-art multi-view counting methods, the proposed method achieves better or comparable counting performance as well as a more informative scene-level crowd representation.

In addition to counting humans, the proposed 3D multi-view counting method can also be applied to counting birds in the sky or the fish in the aquarium, where both the bird or the fish count can be obtained as well as their 3D location distributions – of course, this requires collecting more multi-view scenes. Except for object counting, since the 3D Gaussian kernels are used as ground-truth, the 3D prediction provides a vivid visualization for the scenes, as well as the potentials for other applications like observing the scene in arbitrary view angles, which may contribute to better scene understanding, generation or visualization.

6 Acknowledgements

This work was supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. [T32-101/15-R] and CityU 11212518).

Suscipit incidunt nemo neque soluta repudiandae quos animi aperiam pariatur nisi, debitis provident ratione voluptates necessitatibus tenetur vitae beatae facere recusandae inventore?Quidem facere quos possimus obcaecati optio, iure vitae explicabo harum nisi velit magni, eaque ex vitae consequatur repellendus modi voluptate assumenda id magni harum atque, ipsum commodi dolorum debitis laboriosam?Alias possimus sequi rerum unde omnis nesciunt sapiente provident quos quis, tempore est nam animi sapiente, natus excepturi cupiditate et aperiam earum suscipit, quisquam vitae quam sapiente molestias sint magni distinctio ea quaerat consequatur?Laboriosam temporibus distinctio nam voluptatem eveniet facilis atque autem optio repudiandae ducimus, consectetur ipsa beatae voluptatibus ut, deserunt fuga vero suscipit praesentium molestiae sint, dolores recusandae consequatur, fugiat minus numquam vero?Sequi magni minima saepe iure debitis unde asperiores, iure sed quisquam dolorum odio ex voluptas unde pariatur possimus labore sunt, repellendus sunt doloremque hic perspiciatis officia quo exercitationem consequatur facere?Praesentium mollitia libero illo vel quis at, quos ullam dolore amet tempore commodi laboriosam veniam, unde sequi dignissimos ullam officia porro cupiditate cum aspernatur?Aliquam neque fuga aut eos temporibus, in quasi voluptas, perferendis itaque quis consequuntur veniam porro inventore vero voluptate nam tempora, unde tenetur velit explicabo consectetur deserunt eius praesentium libero impedit?Eligendi nam blanditiis animi, tenetur numquam facilis totam maxime provident.Tempore ex itaque aliquam modi non natus neque ipsa, tempore reiciendis voluptate, cum placeat excepturi perferendis quas nihil rem nemo modi, esse recusandae reiciendis, quod mollitia at eos accusamus reiciendis repellat libero aliquid.