Manifold-Based Visual Object Counting

Yi Wang, Yuexian Zou, *Senior Member, IEEE*, and Wenwu Wang, *Senior Member, IEEE*

***Abstract*— Visual object counting (VOC) is an emerging area in computer vision which aims to estimate the number of objects of interest in a given image or video. Recently, object density based estimation method is shown to be promising for object counting as well as rough instance localization. However, the performance of this method tends to degrade when dealing with new objects and scenes. To address this limitation, we propose a manifold-based method for visual object counting (M-VOC), based on the manifold assumption that similar image patches share similar object densities. Firstly, the local geometry of a given image patch is represented linearly by its neighbors using a predefined patch training set, and the object density of this given image patch is reconstructed by preserving the local geometry using locally linear embedding. To improve the characterization of local geometry, additional constraints such as sparsity and non-negativity are also considered via regularization, nonlinear mapping, and kernel trick. Compared with the state-of-the-art VOC methods, our proposed M-VOC methods achieve competitive performance on seven benchmark datasets. Experiments verify that the proposed M-VOC methods have several favorable properties, such as robustness to the variation in the size of training dataset and image resolution, as often encountered in real-world VOC applications.**

***Index Terms*— Visual object counting, object density map estimation, manifold-based, locally linear embedding, manifold assumption, kernel method.**

I. Introduction

VISUAL object counting (VOC) is one of the most active research areas in computer vision and signal processing which aims to predict the number of objects in an image or video, and to infer the statistics of the objects in a given scene. This technique can be employed in a number of applications, e.g. cell counting in medical imaging, bird census in wild observation, and crowd monitoring in public areas (see examples shown in Figure 1).

Manuscript received April 9, 2017; revised August 28, 2017, October 21, 2017, December 16, 2017, and January 18, 2018; accepted January 20, 2018. Date of publication January 29, 2018; date of current version April 6, 2018. This work was supported by the Shenzhen Science and Technology Fundamental Research Programs under Grant ZDSYS201703031405467 and Grant JCYJ20160330095814461. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Tolga Tasdizen. *(Corresponding author: Yuexian Zou.)*

Y. Wang and Y. Zou are with the School of Electrical and Computer Engineering, Peking University Shenzhen Graduate School, Shenzhen 518055, China (e-mail: wygamle@pku.edu.cn; zouyx@pkusz.edu.cn).

W. Wang is with the Department of Electrical and Electronic Engineering, University of Surrey, Guildford GU2 7XH, U.K. (e-mail: w.wang@surrey.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2018.2799328

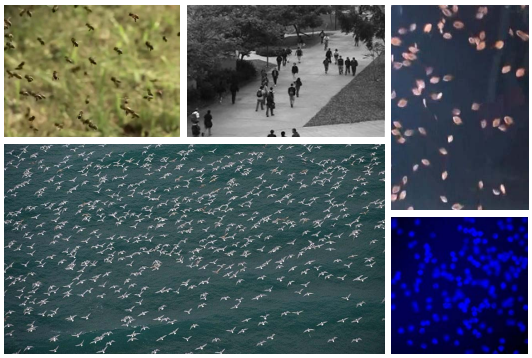


Fig. 1. Illustration of object types: bees (upper left), pedestrians (upper middle), fishes (upper right), seagulls (bottom left) and cells (bottom right).

A. Related Work

Existing VOC methods are approximately categorized into four types: (1) counting by detection; (2) counting by trajectory-clustering; (3) counting by global regression; (4) counting by object density estimation. The counting by detection method has been used in pedestrian counting and it works well when most people in the scene are separated clearly, but its performance degrades significantly when the objects get closer or are occluded by each other [1]–[4]. The counting by trajectory-clustering method [5]–[7] is designed to count crowded moving objects, thus it can only be applied to videos or image sequences for acquiring desired trajectories. Moreover, the clustering process often incurs high computational cost. The counting by global regression method yields fairly good estimation by using swift training and testing procedure, however it relies heavily on feature engineering [8]–[15], and cannot give specific object distribution information.

The counting by object density estimation (DE-VOC) methods, introduced originally in [16], estimate a real-valued density function of pixels in a given image by mapping the local features of the image to its density map [16]–[22]. The DE-VOC methods are usually composed of three common elements: with manually labeled training images, the DE-VOC methods firstly generate the ground truth density map, then extract the local features and finally apply a regression model to learn the mapping between the local features and its corresponding density map. Consequently, the learned regression model can be used to estimate the density map of any given image, and the corresponding object count is calculated as the integral of the density map. Different from other VOC methods [2], [5], [8]–[13], [23]–[25], the DE-VOC methods

yield object density maps that are useful for the analysis of object distributions across the whole image.

In [16], the ground truth object density map is generated by convolving the object location map with a Gaussian kernel. Then, the coded dense SIFT feature is taken to represent the image, and finally a linear regression model is employed to learn the mapping between the features and density maps. The method was shown to be robust to additive local perturbations [16]. This method has been further extended in [22] by integrating the perspective map into the generation of the ground truth density map, and in [19] for efficient implementation by using regression forests instead of linear regression.

Recently, convolutional neural networks (CNN) have been applied to solve the VOC problem [26]–[28]. Compared with conventional DE-VOC methods, the feature engineering process is replaced by feature learning in a supervised manner. One example is presented in [27], which gave the state-of-the-art performance with about 4K manually labeled frames in a 200K pedestrians dataset in 2015. It is noted that the CNN based VOC methods are facilitated by the availability of large scale training data and high performance computational resources e.g. graphical processing units (GPUs). For many real applications, however, only relatively small datasets are available, and this motivates us to develop an effective DE-VOC method with limited training data instead of the CNN based methods with large scale training data.

B. Motivations

The performance of the DE-VOC methods, however, tends to degrade when dealing with new objects and scenes [16], [17]. To address this limitation, in this paper, we propose a novel manifold-based DE-VOC method (M-VOC), where the object density map is estimated from a training dataset, based on the manifold assumption [29], [30] that the neighboring image patches are more likely to share similar density patches while the distant ones are less likely to. This assumption is made based on the observation that the image of objects shares the same information as its density map regarding the location of the objects in space, and recurrent patterns appear everywhere in natural swarm scenes such as crowds and birds.

In our proposed M-VOC, the density map of a given image patch is reconstructed based on its local geometry since the image patches that lie in a manifold share a similar local geometry as the manifold formed by their object density maps. As a result, the VOC problem is converted to the problem of characterizing the local geometry of the given image patch. For this reason, the proposed method is robust against features used and image resolution.

To capture the local geometry of the input image patch, the locally linear embedding (LLE) method, which has been extensively studied in manifold learning [31], [32], is adopted in the proposed M-VOC. The LLE method has been applied to multi-view and cross-modal applications [33]–[36], where multi-modal features are exploited for image retrieval, classification or regression problems. Different from these works, however, the proposed M-VOC method focuses on modelling and deriving the correspondence from images to their density maps. To our knowledge, it is the first time that the LLE method is used in a VOC problem.

To further improve the performance of LLE, additional regularizations, namely, energy, sparsity, and non-negativity constraints are considered. With these regularizations, however, it becomes less trivial to compute the local geometry. To address this limitation, nonlinear mapping based on a kernel method can be incorporated into the proposed M-VOC, which we name as the KM-VOC method. With this method, no regularization terms will be required and the algorithm becomes more tractable. Specific kernels such as the Radial Basis Function (RBF) is used to induce non-negativity and sparsity simultaneously in the local geometry. Although the regularized and kernel versions of LLE have been studied in [37]–[40], they have not been applied to the VOC problem. We show that the kernel and regularized LLE is highly relevant to the VOC problem. The kernel method offers an efficient solution to the regularized LLE, while the regularization on LLE renders desirable properties in the VOC problem such as sparsity and non-negativity.

In addition, to find similar patches more efficiently, instead of using conventional nearest neighbor searching algorithms, a hierarchical searching method is developed which uses a simple tree structure to convert the complexity of the problem from $O(N)$ to $O(\log N)$, where N is the number of samples in the training data. To further improve the computational efficiency, a pre-trained local regression method [41]–[43] is adopted to approximate the desired local geometry in our KM-VOC method, which is able to eliminate the neighborhood search process.

It is worth pointing out that the proposed M-VOC essentially differs from the conventional DE-VOC methods in the following two aspects. The manifold assumption is firstly introduced to solve the VOC problem. In addition, the proposed M-VOC method is a nonparametric approach while the mainstream DE-VOC methods use parametric regression models.

C. Contributions

To make it clear, our contributions in this work are summarized as follows:

1) Based on the manifold assumption for the VOC problem, a novel manifold-based VOC method (M-VOC) has been proposed for generic object counting, by exploiting the similarity in the local geometry between the images and their corresponding density maps.

2) To better characterizing the local geometry, sparse and non-negative representations are also considered via regularizations and nonlinear mapping with kernel trick, which leads to several variants of the proposed method.

3) The local pattern learning and hierarchical searching have been employed to further improve the computational efficiency of the proposed M-VOC method and its variants.

Preliminary results of our work can be found in [44] and [45]. Current work adds to the initial version in several significant aspects. Firstly, more local geometry regularizations have been investigated, and theoretical

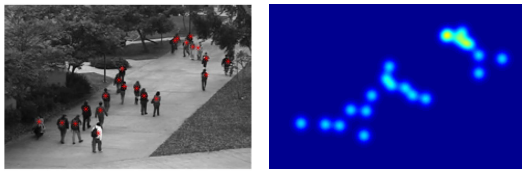


Fig. 2. The generation of ground truth density map. (a) Left: A pedestrian image with user annotations on object locations (red stars); (b) Right: the generated density map (displayed in jet colormap).

analysis and experimental validation are given to illustrate the performance improvement of the proposed method. Secondly, by introducing the kernel method, the original local geometry constraints, such as non-negativity and sparsity can be achieved implicitly, which not only gives a more compact and uniform formulation but also boosts the performance in object counting. Thirdly, our experiments are extended from pedestrian and cell datasets to insect, fish and bird datasets, and substantial new analyses are provided to the initial results as well as to the new experimental results.

D. Paper Organization

The remainder of the paper is organized as follows: Section II presents the idea, formulations, and the algorithmic implementations of our proposed M-VOC method; in Section III, extensive experiments are conducted on benchmark datasets to evaluate the performance of our M-VOC, as compared with several state-of-the-art VOC methods. Lastly, Section IV concludes the paper.

II. Proposed Method and Algorithm

In a conventional DE-VOC method, for a given image $\boldsymbol{X}$, the density map $\boldsymbol{X}_d$ is estimated first before the object counts $c(\boldsymbol{X})$ is computed by taking the integral over $\boldsymbol{X}_d$.

In this section, from a new perspective, we proposed a novel approach to estimate the density map $\boldsymbol{X}_d$ and derive several variants based on how to regularize the local geometry to obtain effective local linear representation and their corresponding solutions. As our method estimates object density using manifold learning techniques under a manifold assumption, it is named as manifold-based visual object counting (M-VOC).

A. The Main Assumption and Key Ideas

Our method is inspired by two key observations. To explain this, in Figures 2 and 3, we show two example images and one produced density map (generated by using the algorithm in [16], more details are given in Section II.B). From Figure 2, it is noted that the image of objects shares the same object location information with its density map in spatial space. In Figure 3, many image patches share similarity in the counting scene, indicating that recurrent patterns are everywhere in natural swarm scenes such as crowds and birds. With these two observations, *we make the manifold assumption in the counting problems: the similar image patches are more likely to share similar density patches while the dissimilar ones are less likely to. Under this assumption, the image patches and their corresponding density patches could be viewed as lying in two manifolds that share a similar local geometry.*

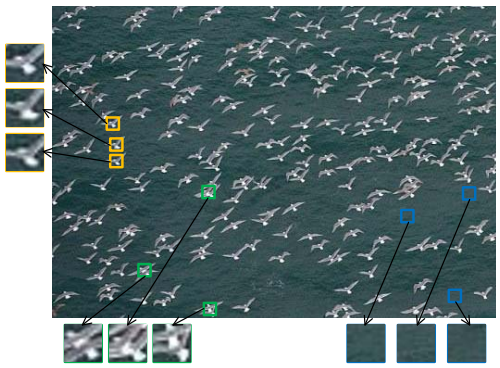


Fig. 3. An illustration about recurrent patterns in the counting scene. The regions marked by the same color share the same pattern.

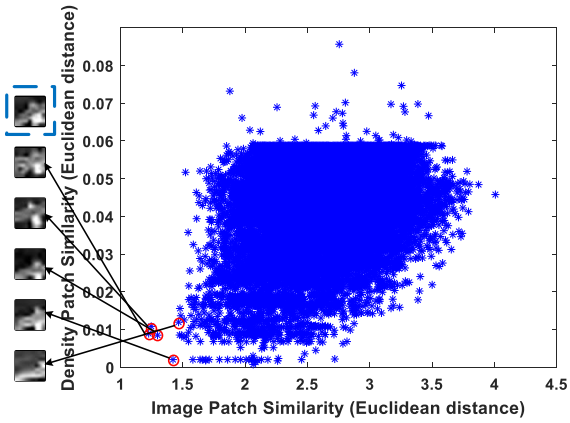


Fig. 4. An illustration of the manifold assumption made in our proposed M-VOC method. The test image patch is shown on the top left marked by blue dash bounding box. Each cross represents the "image patch similarity score obtained from $\boldsymbol{x}$ and $\boldsymbol{y}^i$ versus the "density patch similarity score obtained from $\boldsymbol{x}_d$ and $\boldsymbol{y}_d^i$. The five training patches, which are most similar to the test image patch in terms of the "image patch similarity" measure, are shown on the top left (below the test image patch), whose similarity scores are highlighted with red circles. Here showing the similarities between the input patch and all the training patches is to demonstrate the fact that, although some training patches are most similar to the input image patch, their density patches may not be the ones that are most similar to the density patch of the input image patch.

Let $\boldsymbol{x}$ be the image patch extracted from $\boldsymbol{X}$, while its density patch be $\boldsymbol{x}_d$. Denote the annotated training images as $\boldsymbol{I}^i (i = 1, 2, \ldots, N)$, and the set of image patches as $\boldsymbol{Y} = \{\boldsymbol{y}^1, \boldsymbol{y}^2, \ldots, \boldsymbol{y}^M\}$, where $\boldsymbol{y}^i \in \mathbb{R}^{q_1}$. Accordingly, the set of the density patches of the corresponding image patches is denoted as $\boldsymbol{Y}_d = \{\boldsymbol{y}_d^1, \boldsymbol{y}_d^2, \ldots, \boldsymbol{y}_d^M\}$ where $\boldsymbol{y}_d^i \in \mathbb{R}^{q_2}$ are extracted from $\boldsymbol{I}_d^i (i = 1, 2, \ldots, N)$. The aim of the M-VOC method is to estimate $\boldsymbol{x}_d$ for a given $\boldsymbol{x}$.

With the manifold assumption, $\boldsymbol{x}$ and $\boldsymbol{x}_d$ share the similar local geometry. This means that, if $\boldsymbol{x}$ can be represented by its neighbors in a certain way in order to capture the local geometry, then $\boldsymbol{x}_d$ can also be represented by its neighbors in the same way. The similarity on the local geometry between

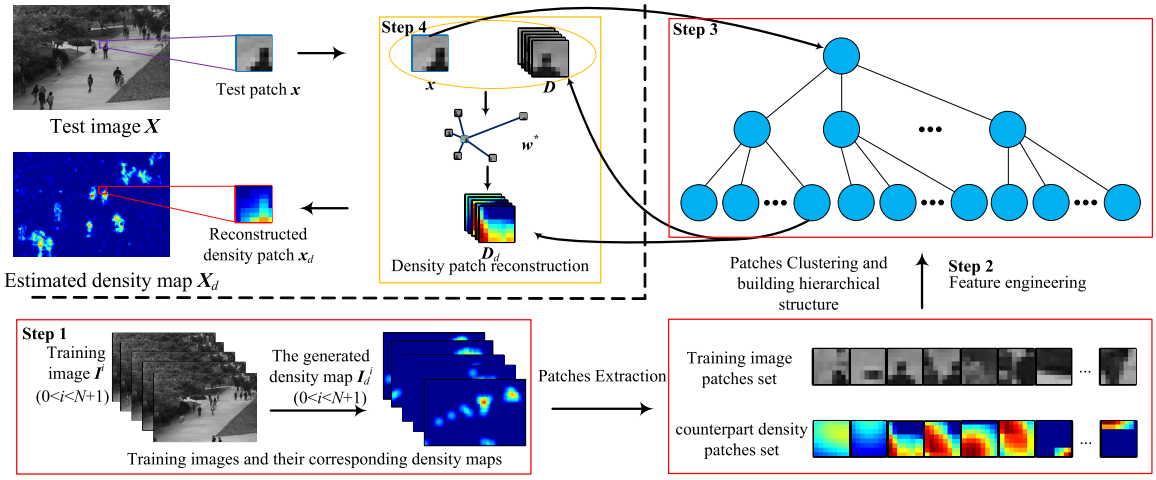


Fig. 5. The pipeline of the proposed manifold-based visual object counting. All the testing procedures are in orange boxes (in the upper left of the figure) while the training ones are in red boxes (the remaining part).

$\boldsymbol{x}$ and $\boldsymbol{x}_d$ can be expressed as:

$$\begin{cases} \boldsymbol{x} = \boldsymbol{D}\boldsymbol{w} \\ \boldsymbol{x}_d = \boldsymbol{D}_d\boldsymbol{w} \end{cases} \tag{1}$$

where $\boldsymbol{D} = [\boldsymbol{y}^{t_1}, \boldsymbol{y}^{t_2}, \ldots, \boldsymbol{y}^{t_T}]$ is the subset formed by the T nearest neighbors of $\boldsymbol{x}$ from $\boldsymbol{Y}$, $\boldsymbol{D}_d = [\boldsymbol{y}_d^{t_1}, \boldsymbol{y}_d^{t_2}, \ldots, \boldsymbol{y}_d^{t_T}]$ is the subset of density patches corresponding to $\boldsymbol{D}$, and $\boldsymbol{w}$ is the weight vector describing the local geometry of $\boldsymbol{x}$ and $\boldsymbol{x}_d$. In theory, $\boldsymbol{w}$ can be jointly computed from $(\boldsymbol{x}, \boldsymbol{D})$ and $(\boldsymbol{x}_d, \boldsymbol{D}_d)$. In practice, however, $\boldsymbol{x}_d$ is unknown and needs to be predicted from $\boldsymbol{x}$. As a result, it is not a trivial task, if not impossible, to estimate $\boldsymbol{w}$ jointly from $(\boldsymbol{x}, \boldsymbol{D})$ and $(\boldsymbol{x}_d, \boldsymbol{D}_d)$.

To further clarify the manifold assumption, we illustrate the relation of the density patch similarity to image patch similarity using a plot. Figure 4 is generated using the Seagull dataset [46]. First, we choose a cropped test image patch $\boldsymbol{x}$ of size at 9×9 pixels, as shown on the top left side of the figure (highlighted with blue dash bounding box), and 23180 image patches $\boldsymbol{y}^i$, $(i = 1, 2, \ldots, 23180)$ from a training set. We measure the image patch similarity between $\boldsymbol{x}$ and $\boldsymbol{y}^i$ by their Euclidean distance as $s^i = ||\boldsymbol{x} - \boldsymbol{y}^i||_2$, $i = 1, \ldots, 23180$, which is shown along the horizontal axis of the figure. Define the density patch of $\boldsymbol{x}$ and $\boldsymbol{y}^i$ as $\boldsymbol{x}_d$ and $\boldsymbol{y}_d^i$, respectively. The density patch similarity between $\boldsymbol{x}_d$ and $\boldsymbol{y}_d^i$ is also measured by their Euclidean distance denoted as $ds^i = ||\boldsymbol{x}_d - \boldsymbol{y}_d^i||_2$, $i = 1, \ldots, 23180$, which is shown along the vertical axis. A lower Euclidean distance indicates a higher similarity. In this figure, we show 23180 cross points (s^i, ds^i), $i = 1, \ldots, 23180$. In addition, we highlight five crosses using red circles at the bottom left whose s^i values are the five highest among the 23180 crosses. Carefully examining these five crosses, we get the following paired values of (s^i, ds^i): $(1.2261, 0.0089)$, $(1.2467, 0.0108)$, $(1.2973, 0.0085)$, $(1.4215, 0.0020)$ and $(1.4661, 0.0119)$, respectively. It is noted that ds^i for these five points ranges from 0.002 to 0.011 while s^i ranges from 1.2 to 1.47. This experimental result shows that similar image patches tend to give similar density patches, and vice versa. This validates the manifold assumption that we have made.

B. The Pipeline of the Proposed M-VOC Method

The whole pipeline of our proposed M-VOC is given in Figure 5 which contains four key steps as follows: 1) the ground truth density map generation (in the bottom-left corner of Figure 5); 2) feature engineering; 3) building search structure (in the top-right corner of Figure 5); 4) density map reconstruction (in top-left corner of Figure 5). The details of each step will be discussed in the following subsection. The main novel contributions of our work are in steps 3 and 4, while in steps 1 and 2, existing techniques are used.

1) The Generation of the Ground Truth Density Maps: There are several methods that have been proposed to estimate the density map [16], [17], which will be reviewed briefly for presentation clarity. Usually, the annotations by users on object locations are discrete 2D points in the image as shown in Figure 2(a). In order to make the object locations change continuously, the object location map is kernelized to obtain a smoothed object distribution [41]. Suppose a set of N manually annotated images $\boldsymbol{I}^1, \boldsymbol{I}^2, \ldots, \boldsymbol{I}^N$ are pre-allocated. Then, the ground truth density maps $\boldsymbol{I}_d^i$ are usually defined as a sum of 2D kernels of the object locations [16], as:

$$\boldsymbol{I}_d^i(z) = \sum\nolimits_{U \in \boldsymbol{U}^i} \mathcal{N}(z; U, \sigma^2 \boldsymbol{1}_{2\times 2}) \tag{2}$$

where $\boldsymbol{I}_d$ indicates the ground truth density map of $\boldsymbol{I}$, z is the pixel index of image $\boldsymbol{I}^i$, i is the image index, U is the user-annotated dot, and $\boldsymbol{U}^i$ is a 2D points set marking all object locations in $\boldsymbol{I}^i$. Moreover, $\mathcal{N}$ is the normalized 2D Gaussian kernel function. σ^2 is the variance of $\mathcal{N}$ for smoothing the local distribution, and is set according to the size of objects (approximately 1/2 size of objects). One example of the generated ground truth density map can be found in Figure 2(b).

With $\boldsymbol{I}_d^i$, the object count $c(\boldsymbol{I}^i)$ is given by:

$$c(\boldsymbol{I}^i) = \sum\nolimits_{z \in \boldsymbol{I}_d^i} \boldsymbol{I}_d^i(z) \tag{3}$$

2) Feature Engineering: As discussed above, the existing DE-VOC methods require sophisticated hand-crafted or learned local features from images. For

generalization purpose, simple or less feature engineering is desired since feature engineering is usually application and scene dependent. Here, we seek methods to preserve object distribution information. Our preliminary research shows that raw image data feature is an appropriate candidate. *To increase sampling densities in feature space and reduce the computational burden*, the raw data features in patch form are centralized, normalized and dimension-reduced by PCA. However, we also considered engineered features in our experiments in Section III. B. 7.

3) Building Searching Structure: The realization of locality (i.e. the construction of $\boldsymbol{D}$ in (1)) is usually achieved by searching the whole example space, which is time-consuming even with advanced search structure like KD-Tree [47]. To accelerate the M-VOC in its testing phase, we compromise its training time with a hierarchical search structure whose nodes are generated by clustering, similar to the idea used in [48]. In our study, a two-layer hierarchical search scheme is employed. Without loss of generality, assume $\boldsymbol{Y}$ has K clusters. Then, there are $\sqrt{K}$ nodes in the first layer, which are the centroids of the $\sqrt{K}$ clusters of $\boldsymbol{Y}$ obtained by the K-Means algorithm. For the second layer, each node in the first layer has $\sqrt{K}$ children nodes, which are the centroids of the $\sqrt{K}$ clusters of the image patches from $\boldsymbol{Y}$ assigned to their feature nodes.

4) Density Map Reconstruction: In this subsection, different from the aforementioned mainstream DE-VOC methods that use the regression model to compute $\boldsymbol{x}_d$ from $\boldsymbol{x}$, or $\boldsymbol{X}_d$ from $\boldsymbol{X}$, *we present a nonparametric method based on the manifold assumption* to learn the weight vector $\boldsymbol{w}$ for $\boldsymbol{x}$ firstly, then use $\boldsymbol{w}$ to estimate $\boldsymbol{x}_d$.

Let $\mathcal{J}(\boldsymbol{w}|\boldsymbol{x}, \boldsymbol{D})$ denote the cost function for computing $\boldsymbol{w}$ based on $\boldsymbol{x}$ and $\boldsymbol{D}$. For each input patch $\boldsymbol{x}$ extracted from the test image $\boldsymbol{X}$, $\boldsymbol{w}$ is obtained by solving the following optimization problem, expressed as

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \ \mathcal{J}(\boldsymbol{w}|\boldsymbol{x}, \boldsymbol{D}) \quad s.t. \ \mathbf{1}^{\mathrm{T}}\boldsymbol{w} = 1 \tag{4}$$

Then, the estimation of $\boldsymbol{x}_d$ can be computed by:

$$\boldsymbol{x}_d \cong \boldsymbol{D}_d\boldsymbol{w}^* \tag{5}$$

Finally, $\boldsymbol{x}_d$ is put into $\boldsymbol{X}_d$ at the same position as $\boldsymbol{x}$ in $\boldsymbol{X}$. After each patch in $\boldsymbol{X}$ is processed, the density map $\boldsymbol{X}_d$ is estimated, and the count of $\boldsymbol{X}$ is obtained as $c(\boldsymbol{X}) = \sum_{z\in\boldsymbol{X}_d} \boldsymbol{X}_d(z)$.

C. Proposed M-VOC Algorithm

From (1), the key is to minimize the linear reconstruction error $\mathcal{J}(\boldsymbol{w}|\boldsymbol{x}, \boldsymbol{D}) = \|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{w}\|_2^2$ between $\boldsymbol{x}$ and $\boldsymbol{D}\boldsymbol{w}$. Hence, the solution of $\boldsymbol{w}$ is expressed as

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \ \|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{w}\|_2^2 \quad s.t. \ \mathbf{1}^{\mathrm{T}}\boldsymbol{w} = 1 \tag{6}$$

It is noted that (6) is a standard least squares problem, therefore, if $\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}$ is positive definite, $\boldsymbol{w}$ can be solved efficiently as:

$$\boldsymbol{w}^* = \frac{1}{Z}(\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D})^{-1}\boldsymbol{D}^{\mathrm{T}}\boldsymbol{x} \tag{7}$$

where Z is a normalization factor. The M-VOC method using (7) for computing $\boldsymbol{w}$ is termed as M-VOC(LS).

However, computing the local weights using (7) is unstable when $q_1 > T$, because $\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}$ is not positive definite under this circumstance. Hence, some regularizations are introduced as follows to achieve more reliable local linear representation.

1) Energy: To produce more stable local weights, $\boldsymbol{w}$ can be constrained by its energy, indicating that the possible $\boldsymbol{w}$ will be limited [41].

2) Sparsity: The performance of M-VOC is often affected by the neighborhood size T. Specifically, if T is too small, the neighbors selected are not enough to characterize the local geometry; on the contrary, the neighbors with different geometries tend to be selected, as a result, M-VOC fails to characterize the local geometry. Clearly, a preset T will lead to unstable performance of the M-VOC for different VOC applications.

To address this problem, inspired by the properties of sparsity and its applications in manifold learning [11], [18], [19], we improve the model in (6) by imposing the locality and sparsity constraints simultaneously. This encourages as few neighbors of $\boldsymbol{x}$ to be selected as possible with the same or similar geometry in feature space. Through the improved model, the local geometry can be learned properly, and as a result, setting T becomes unnecessary.

3) Non-Negativity: In (6), due to the fact that $\mathbf{1}^{\mathrm{T}}\boldsymbol{w} = 1$, applying the non-negativity constraint on $\boldsymbol{w}$ will lead to a convex combination of the most similar training image patches or density patches. Thus, the reconstructed input image patch $\boldsymbol{D}\boldsymbol{w}^*$ is the one obtained using the most similar training image patches. Further, when the manifold assumption holds (i.e. the local geometry between image patches and that between density patches are similar), the estimated input density patch $\boldsymbol{D}_d\boldsymbol{w}^*$ is also the interpolated one based on the used training density patches. As a result, both the reconstructed input image patch and the estimated density patch are not novel to the training image patches and density patches. As observed in our experiments, this will improve the counting performance, since only the known image patch space and density patch space are used for inferring the density patch of the input image patch. In addition, the non-negativity constraint helps to improve the sparsity of $\boldsymbol{w}$ [49], as shown in Figure 6 (b). From Figure 6 (a) and (c), we can see that, without the non-negativity constraint, some of the local weights obtained by the optimization become negative. Incorporating the non-negativity constraint, we obtain non-negative weights as shown in Figure 6 (b), which are also more sparse than those in Figure 6 (a). This helps to improve the counting accuracy as observed empirically in our experiments.

4) Locality: As $\boldsymbol{D}$ used for reconstructing $\boldsymbol{x}$ is chosen from the neighborhood of $\boldsymbol{x}$, locality is assumed implicitly.

Based on the aforementioned four constraints, the optimal $\boldsymbol{w}$ is reformulated from (6) as:

$$\begin{aligned}\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \ &\|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{w}\|_2^2 + \lambda_1\|\boldsymbol{w}\|_2^2 + \lambda_2\|\boldsymbol{w}\|_1 + \lambda_3(\boldsymbol{w} - \mathbf{0}) \\ &s.t. \ \mathbf{1}^{\mathrm{T}}\boldsymbol{w} = 1 \ and \ \lambda_1, \lambda_2, \lambda_3 \geq 0\end{aligned} \tag{8}$$

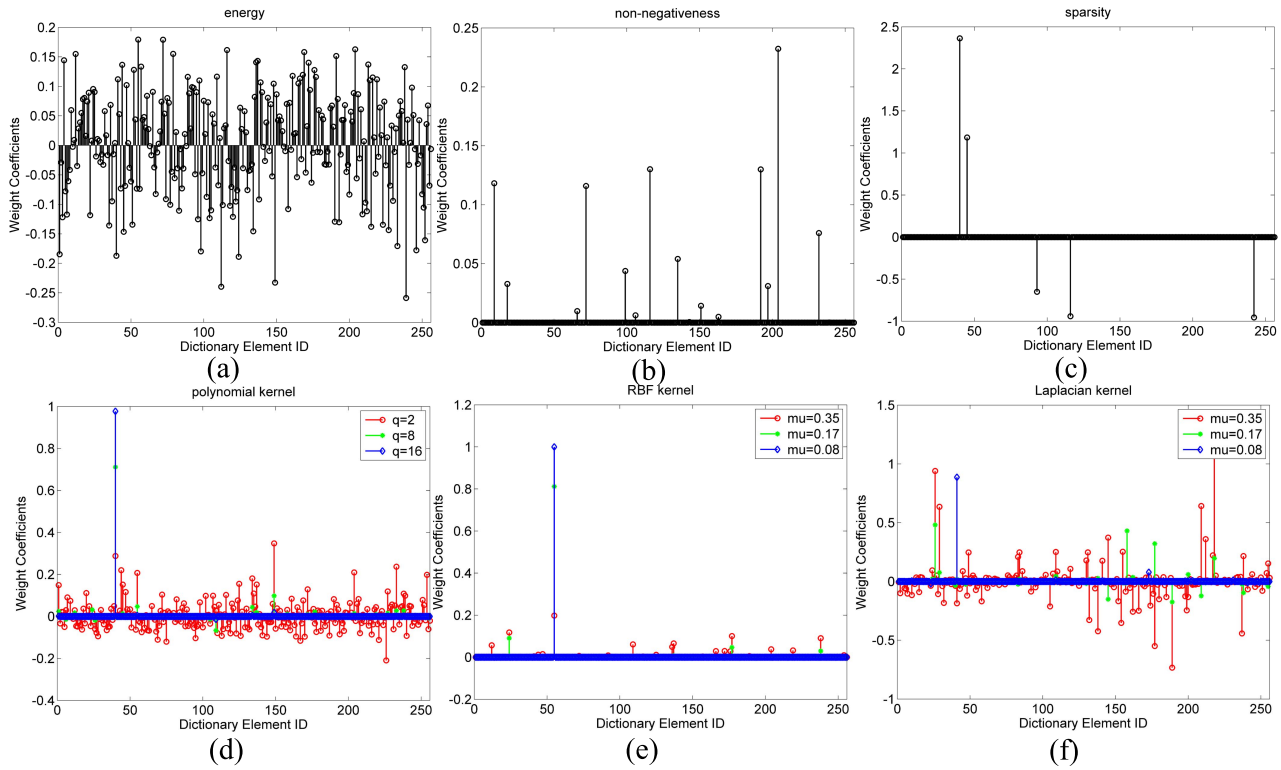


Fig. 6. The distributions of the weight coefficients ($\boldsymbol{w}^*$) obtained by using different kernels. (a) Linear kernel (M-VOC with energy constraint); (b) linear kernel (M-VOC with nonnegativity constraint); (c) linear kernel (M-VOC with sparsity constraint); (d) polynomial kernel ($q = 2$, 8, or 16); (e) RBF kernel ($\mu = 0.35$, 0.17, or 0.08); (d) Laplacian kernel ($\mu = 0.35$, 0.17, or 0.08).

where λ_1, λ_2, and λ_3 are regularization parameters. The second term enforces $\boldsymbol{w}$ with low energy while the third term enforces the sparsity for selecting potential candidates. The fourth term ensures that $\boldsymbol{w}$ is positive. The sparsity constraint eliminates the choice of the neighborhood size by using neighbors as few as possible which essentially favors the neighbors with similar structure [50], [51]. With the joint constraints on energy, sparsity, non-negativity and locality, the selected neighboring candidates tend to share the same or similar geometry.

To get more insights from (8), by setting different λ_1, λ_2, and λ_3, three variants are obtained as follows:

1) Let $\lambda_2 = 0$ and $\lambda_3 = 0$, then (8) is reduced to

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \ \|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{w}\|_2^2 + \lambda_1 \|\boldsymbol{w}\|_2^2 \quad s.t.\ \mathbf{1}^{\mathrm{T}}\boldsymbol{w} = 1 \ and \ \lambda_1 \geq 0 \tag{9}$$

With $q_1 > T$, (9) is of a constrained least squares form and it has an analytical solution as:

$$\boldsymbol{w}^* = \frac{1}{Z}(\boldsymbol{D}^{\mathrm{T}}\boldsymbol{D} + \lambda_1 \boldsymbol{I})^{-1}\boldsymbol{D}^{\mathrm{T}}\boldsymbol{x} \tag{10}$$

In this study, M-VOC using (10) for computing $\boldsymbol{w}$ is termed as M-VOC(e).

2) Let $\lambda_1 = 0$ and $\lambda_3 = 0$, then (8) becomes

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \ \|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{w}\|_2^2 + \lambda_2 \|\boldsymbol{w}\|_1 \quad s.t.\ \mathbf{1}^{\mathrm{T}}\boldsymbol{w} = 1 \ and \ \lambda_2 \geq 0 \tag{11}$$

Equation (11) can be solved by Lasso or the basis pursuit algorithms [52]. The sparsity yielded by the l_1-norm constraint avoids the choice of T since (11) guarantees that the smallest T is used. Similarly, M-VOC using (11) for computing $\boldsymbol{w}$ is termed as M-VOC(s).

3) Let $\lambda_1 = 0$ and $\lambda_2 = 0$, then energy and sparsity will have no effects on $\boldsymbol{w}$, so (8) gives:

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \ \|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{w}\|_2^2 \quad s.t.\ \mathbf{1}^{\mathrm{T}}\boldsymbol{w} = 1 \ and \ \boldsymbol{w} \geq \mathbf{0} \tag{12}$$

Equation (12) is actually a non-negative least squares (NNLS) formulation, which can be solved effectively by quadratic programming (QP) tools.

The introduction of non-negativity to the local geometry also induces sparsity according to [49] and [53], which will be shown in Section III.B. The M-VOC method using (12) for computing $\boldsymbol{w}$ is termed as M-VOC (nn).

We have just given formulations on how to estimate $\boldsymbol{w}$ (and then $\boldsymbol{x}_d$) when $\boldsymbol{x}$ is given. Therefore, the way to estimate $\boldsymbol{X}_d$ from the whole image $\boldsymbol{X}$ is summarized in Algorithm 1.

D. Proposed KM-VOC Algorithm

Image patches contain numerous variations like shapes and textures, and a linear representation as discussed in the above section may not be able to fully capture their underlying intrinsic relationship. Here *we firstly incorporate nonlinear mapping into the modeling of the local geometry in M-VOC, and then apply a kernel method to make it tractable.* This kernel based M-VOC method is termed as KM-VOC.

Algorithm 1 The M-VOC Method

Input: Test image $\boldsymbol{X}$, and training examples set $\boldsymbol{Y}$ and $\boldsymbol{Y}_d$
Output: Density map $\boldsymbol{X}_d$, the estimated count $c(\boldsymbol{X})$

1: **for** Each input patch $\boldsymbol{x}^i$ (the i_{th} patch) extracted from the test image $\boldsymbol{X}$ **do**
2: Find $\boldsymbol{D} = [\boldsymbol{y}^{t_1}, \boldsymbol{y}^{t_2}, ..., \boldsymbol{y}^{t_T}]$, $\boldsymbol{D} \subseteq \boldsymbol{Y}$, whose elements are the most similar T patches compared with $\boldsymbol{x}^i$ (The method to determine $\boldsymbol{D}$ is given in Section II.B.3). The set of the counterpart density maps $\boldsymbol{D}_d = [\boldsymbol{y}_d^{t_1}, \boldsymbol{y}_d^{t_2}, ..., \boldsymbol{y}_d^{t_T}]$ is formed from $\boldsymbol{Y}_d$ according to $\boldsymbol{D}$.
3: Compute local geometry $\boldsymbol{w}^*$ by (7), (10), (11), or (12).
4: Compute the density map patch: $\boldsymbol{x}_d^i = \boldsymbol{D}_d \boldsymbol{w}$. Put $\boldsymbol{x}_d^i$ into $\boldsymbol{X}_d$ at the same position as $\boldsymbol{x}^i$ in $\boldsymbol{X}$.
5: **end for**
6: Get the estimated density map of $\boldsymbol{X}$: $\boldsymbol{X}_d$, and the estimated count of $\boldsymbol{X}$ is given by $c(\boldsymbol{X}) = \sum_{z \in \boldsymbol{X}_d} \boldsymbol{X}_d(z)$

In KM-VOC, a nonlinear mapping Φ is introduced to project $\boldsymbol{x}$ to a much higher dimension as:

$$\Phi : \boldsymbol{x} \mapsto \Phi(\boldsymbol{x}) \in \mathcal{F} \tag{13}$$

where $\Phi(x) \in \mathbb{R}^f$ with $f \gg q_1$. The LLE is then applied to $\Phi(x)$ instead of $\boldsymbol{x}$, in a similar way, as:

$$\begin{aligned} \boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \ & \|\Phi(\boldsymbol{x}) - \Phi(\boldsymbol{D})\boldsymbol{w}\|_2^2 + \lambda \|\boldsymbol{w}\|_2^2 \\ s.t. \ & \mathbf{1}^{\mathrm{T}} \boldsymbol{w} = 1 \ and \ \lambda \geq 0 \end{aligned} \tag{14}$$

where $\Phi(\boldsymbol{D}) = [\Phi(\boldsymbol{y}^{t_1}), \Phi(\boldsymbol{y}^{t_2}), \ldots, \Phi(\boldsymbol{y}^{t_T})]$. Hence, its close-form solution is derived as:

$$\boldsymbol{w}^* = \frac{1}{Z}(\Phi(\boldsymbol{D})^{\mathrm{T}}\Phi(\boldsymbol{D}) + \lambda \boldsymbol{I})^{-1}\Phi(\boldsymbol{D})^{\mathrm{T}}\Phi(\boldsymbol{x}) \tag{15}$$

In $\mathcal{F}$, the linear reconstruction is much easier to achieve than that in the original feature space spanned by image patches according to Cover's theorem [41], implying that there is a high possibility that $\boldsymbol{w}$ in (15) is more effective than $\boldsymbol{w}$ in (7) or (10) on linear reconstruction. We believe that using proper nonlinear mapping function, the local geometry in $\mathcal{F}$ can be better characterized since the image patches that share the similar counts and structures tend to live closer in these spaces.

As studied in the literature [30], [41], there is no need to access the feature $\Phi(\boldsymbol{x})$ or $\Phi(\boldsymbol{D})$ as only the linear correlations between them matter. Let's define a kernel $k(,)$ corresponding to the nonlinear mapping Φ, so (15) is derived as follows:

$$\boldsymbol{w}^* = \frac{1}{Z}(\boldsymbol{G} + \lambda \boldsymbol{I})^{-1} k(\boldsymbol{D}, \boldsymbol{x}) \tag{16}$$

where $\boldsymbol{G}$ is the Gram matrix (which is semi-positive) of $\boldsymbol{D}$, and $\boldsymbol{G}_{i,j} = \Phi(\boldsymbol{y}^{t_i})^{\mathrm{T}}\Phi(\boldsymbol{y}^{t_j})$. $k(\boldsymbol{D}, \boldsymbol{x})$ is the kernel between $\boldsymbol{D}$ and $\boldsymbol{x}$. Obviously, the computation of $\boldsymbol{w}^*$ in (10) is a special case of (16) where $k(.)$ is a linear kernel, indicating $\boldsymbol{G} = \boldsymbol{D}^{\mathrm{T}}\boldsymbol{D}$ and $k(\boldsymbol{D}, \boldsymbol{x}) = \boldsymbol{D}^{\mathrm{T}}\boldsymbol{x}$.

To further explore the property of $\boldsymbol{w}$ obtained from (16), an experiment using the UCSD pedestrian data [9] is conducted. Firstly, a testing patch $\boldsymbol{v} \in \mathbb{R}^{16}$ (in column vector form) is extracted. Then, 256 nearest neighbors of $\boldsymbol{v}$ are extracted from the training set. After that the optimal weight vector ($\boldsymbol{w}^*$) used for constructing $\boldsymbol{v}$ is obtained by solving (10), (11), (12), (16) with the polynomial kernel, (16) with the RBF kernel and (16) with Laplacian kernel, respectively. The visualization of $\boldsymbol{w}^*$ is given in Figure 6 (a-f), respectively. In this experiment, λ is all set to $1e-3$.

It can be observed from Figure 6 that $\boldsymbol{w}^*$ computed by both M-VOC and KM-VOC in (10) (e.g. subplot 6 (a)) contains some negative values, however, with proper setting of the kernel parameters, KM-VOC methods (e.g. subplots 6 (d-f)) can potentially improve the non-negativity of $\boldsymbol{w}^*$. For example, with the increase of q in KM-VOC with polynomial kernel, or the decrease of μ in KM-VOC with RBF or Laplacian kernel, the negative values in $\boldsymbol{w}^*$ become close to zero.

Moreover, from Figure 6 (d-f), we can see that the KM-VOC method yields more sparse $\boldsymbol{w}^*$ as compared with that of M-VOC(e) shown in Figure 6 (a). These results indicate that, for the KM-VOC method, few neighboring vectors are used to reconstruct $\boldsymbol{v}$, which implies implicitly the sparsity property of $\boldsymbol{w}^*$. In principle, the sparseness of $\boldsymbol{w}^*$ from (16) may come from the property of the kernel function. For example, for the RBF kernel function, the exponential change in the Euclidean distance between feature vectors ensures that the majority of the weight coefficients in $\boldsymbol{D}$ approach zero unless they live as close as they are in the given range.

For KM-VOC, to further reduce the computational cost, another method termed as anchored neighborhood regression [42], [43] is employed.

In KM-VOC, $\boldsymbol{w}^*$ is obtained by (16). Substituting it into the density patch reconstruction procedure in (1), $\boldsymbol{x}_d$ can be reconstructed as:

$$\boldsymbol{x}_d \cong \boldsymbol{E} k(\boldsymbol{D}, \boldsymbol{x}) \tag{17}$$

where $\boldsymbol{E} = \boldsymbol{D}_d(\boldsymbol{G} + \lambda \boldsymbol{I})^{-1}$ is the embedding matrix computed from the neighborhood of $\Phi(\boldsymbol{x})$, and the image patches in $\boldsymbol{D}$ are the local examples in the neighborhood of $\Phi(\boldsymbol{x})$.

It is observed that the number of distinguishable distribution patterns of the objects (neighborhoods) is limited. Therefore, their embedding matrices and local examples can be computed in advance and stored for later density patch reconstruction. More specifically, suppose the neighborhoods set is defined as $\{\boldsymbol{C}^i\}_{i=1,2,...,K}$, which is the clustering result of $\boldsymbol{Y}$. Hence the counterpart density maps cluster $\boldsymbol{C}_d^i$ is produced by putting the density patches together according to the index set of the corresponding elements in $\boldsymbol{C}^i$. With $\boldsymbol{C}^i$ and $\boldsymbol{C}_d^i$, the embedding matrix can be computed as

$$\boldsymbol{E}^i = \boldsymbol{C}_d^i(\boldsymbol{G}' + \lambda \boldsymbol{I})^{-1} \tag{18}$$

where $\boldsymbol{G}'$ is the gram matrix of $\boldsymbol{C}^i$. Suppose $\boldsymbol{C}^i = \{\boldsymbol{y}^{c_1}, \boldsymbol{y}^{c_2}, \ldots, \boldsymbol{y}^{c_{t'}}\}$ $w.r.t.$ $t' = |\boldsymbol{C}^i|$, where $|.|$ counts the number of elements in $\boldsymbol{C}^i$. Then, $\boldsymbol{G}'_{ij} = \Phi(\boldsymbol{y}^{c_i})^{\mathrm{T}}\Phi(\boldsymbol{y}^{c_j}) = k(\boldsymbol{y}^{c_i}, \boldsymbol{y}^{c_j})$.

For a patch $\boldsymbol{x}$, we need to determine its neighborhood. In this study, we measure the difference between $\boldsymbol{x}$ and the anchored examples $\bar{\boldsymbol{C}}^i$ (the centroid of $\boldsymbol{C}^i$):

$$i^* = \arg\min_{1 \leq i \leq K} \ \mathrm{dist}(\bar{\boldsymbol{C}}^i, \boldsymbol{x}) \tag{19}$$

where i^* is the index of the selected neighborhood. For visualizing this concept, some anchored examples are given

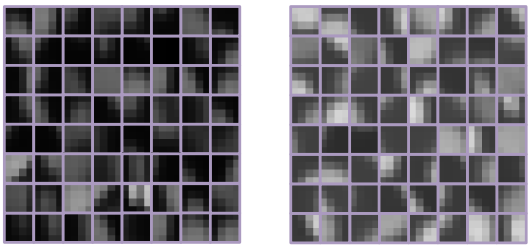


Fig. 7. The partial centroids of the clusters on the Mall (displayed in foreground feature) and Fish (displayed in gray channel) dataset. The patch size is 8×8 and the number of clusters K is set to 256.

in Figure 7. dist$(\cdot)$ is the distance metric and Euclidean distance is used here.

Noteworthy is that the number of examples in neighborhoods $\boldsymbol{C}^i$ is unequal. Some neighborhoods contain more examples which possibly exceed the requirement for well-sampling. Thus, to save computation without comprise on counting performance, $\boldsymbol{C}^i$ will be re-sampled if its size exceeds l. Specifically, the following sampling strategy is taken: for $\boldsymbol{C}^i$ $w.r.t.$ $|\boldsymbol{C}^i| > l$, it will be clustered into l segments as $\{\boldsymbol{C}'^i_{t_1}, \boldsymbol{C}'^i_{t_2}, \ldots, \boldsymbol{C}'^i_{t_l}\}$. So $\boldsymbol{C}^i$ $w.r.t.$ $|\boldsymbol{C}^i| > l$ will be substituted by $\{\bar{\boldsymbol{C}}'^i_{t_1}, \bar{\boldsymbol{C}}'^i_{t_2}, \ldots, \bar{\boldsymbol{C}}'^i_{t_l}\}$, where $\bar{\boldsymbol{C}}'^i_{t_j}$ is the centroid of the $\boldsymbol{C}'^i_{t_j}$. The $\boldsymbol{C}^i_d$ $w.r.t.$ $|\boldsymbol{C}^i_d| > l$ will be updated in the same way.

Testing Phase of KM-VOC: There are two stages in KM-VOC testing phase.

- First, the image patch $\boldsymbol{x}$ extracted from the test image $\boldsymbol{X}$ is assigned to a neighborhood $\boldsymbol{C}^j$ using (19).
- Second, the density patch $\boldsymbol{x}_d$ of $\boldsymbol{x}$ is reconstructed by embedding matrix $\boldsymbol{E}^j$ of that pattern and similarity measure matrix $k(\cdot, \boldsymbol{x})$ corresponding to $\boldsymbol{C}^j$ using (17).

E. Time Complexity Analysis of M-VOC and KM-VOC

In this section, we analyze the time complexity of the proposed algorithms. We focus on the testing phase. Assume that the input testing image $\boldsymbol{X}$ is of size $width \times height$ (or called problem size), the patch size is set to $\sqrt{q_1} \times \sqrt{q_1}$, and the overlap between the neighboring patches is set to half of the patch size as $\frac{\sqrt{q_1}}{2}$. When $\boldsymbol{X}$ is given as input to the algorithm, it is cut into $\frac{width}{\sqrt{q_1}} \times \frac{height}{\sqrt{q_1}}$ overlapping patches. The number of clusters of the salient patterns is K and in each cluster, only l anchors are picked or generated as examples.

1) M-VOC: Due to the use of a two layer hierarchical search structure, for each input image patch, the search for the T nearest neighbors costs $O(Tl + 2\sqrt{K})$. Then the density map reconstruction process in (7) costs $O(T^3)$. Thus, for every testing image patch, this costs $O(Tl + 2\sqrt{K} + T^3)$. In total, the testing phase amounts to a cost of $O(\frac{width \times height}{q_1}) \times O(Tl + 2\sqrt{K} + T^3) = O(\frac{Tl + 2\sqrt{K} + T^3}{q_1} width \times height)$ for the whole image $\boldsymbol{X}$.

2) KM-VOC: For each image patch, the classification stage will cost $O(2\sqrt{K})$ due to the use of a two-layer hierarchical search structure. During the reconstruction stage, the computation of $k(\bar{\boldsymbol{C}}^{t^*}, \boldsymbol{x}_{d,ij})$ costs $O(l)$, then the reconstruction of density map $\boldsymbol{x}_{d,ij} = \boldsymbol{E}^{t^*} k(\bar{\boldsymbol{C}}^{t^*}, \boldsymbol{x}_{d,ij})$ costs $O(q_1 \times l \times l)$, since $\boldsymbol{E} \in \mathbb{R}^{q_1 \times l}$ and $k(\bar{\boldsymbol{C}}^{t^*}, \boldsymbol{x}_{d,ij}) \in \mathbb{R}^{l \times l}$. Thus the whole reconstruction stage will cost $O(l) + O(q_1 \times l \times l) = O(q_1\, l^2)$. Combining the two stages, the testing phase costs $O(\frac{width \times height}{q_1}) \times O(\sqrt{K}) + O(\frac{width \times height}{q_1}) \times O(q_1\, l^2) = O((\frac{\sqrt{K}}{q_1} + l^2) width \times height)$ for the whole image $\boldsymbol{X}$.

Without the salient patterns and hierarchical search structure, for M-VOC, the search for the T nearest neighbors costs $O(TN)$, where N is the number of training patches. Since $N \ggg T$, for every testing image patch, the cost is $O(TN)$. In total, the testing phase has a cost $O(\frac{width \times height}{q_1}) \times O(TN) = O(\frac{TN}{q_1} width \times height)$ for the whole image $\boldsymbol{X}$.

By employing the hierarchical search structure and salient patterns, the time complexity of M-VOC can be reduced significantly from $O(\frac{TN}{q_1} width \times height)$ to $O(\frac{Tl + 2\sqrt{K} + T^3}{q_1} width \times height)$ of M-VOC and $O((\frac{\sqrt{K}}{q_1} + l^2) width \times height)$ of KM-VOC, since $N \ggg T, K, l, q_1$.

TABLE I
DESCRIPTIONS OF SEVEN DATASETS

Dataset	Amount	Resolution	Object counts	Channel
Bacterial Cell	200	256×256	171 ± 64	RGB
Embryo Cell	11	400×400	98 ± 37	Gray
UCSD	2000	238×158	29 ± 9	Gray
Mall	2000	640×480	33 ± 20	RGB
UCF_CC_50	50	different	1273 ± 957	Gray
Honeybee	118	640×480	28 ± 6	RGB
Fish	129	300×410	59 ± 9	RGB
Seagull	3	624×964	866 ± 107	RGB

III. EXPERIMENTAL RESULTS AND DISCUSSIONS

In our study, seven public datasets are used to evaluate the object counting performance since they have different object types. In the following subsections, the details of the datasets, experimental settings, and performance metrics are introduced first, followed by experimental results and analysis.

A. Datasets, Experimental Settings, and Evaluation Metrics

1) Datasets: In this study, cell [16], bee, fish, bird [46] and pedestrian datasets are used. Their detailed information is summarized in Table I (where in $a \pm b$, a and b represent the mean and the standard deviation respectively). The used pedestrian datasets include UCSD [9], Mall [18] and UCF_CC_50 [54], respectively, which contain crowd in completely different environments. Specifically, the data in UCSD are recorded in outdoor and simple scenes while the data in Mall are recorded inside a shopping mall with complicated surroundings. Moreover, the crowd quantities are both sparse on the UCSD and Mall datasets. In UCF_CC_50, some images contain thousands of people.

2) Experimental Settings of M-VOC: Although counting results obtained by DE-VOC and M-VOC are not very sensitive to the choice of σ in (2), we did not set σ casually. Instead, following the same protocol in [16] and [17], σ in this paper is set according to the size of the objects, as approximately 1/2 size of the objects. Specifically, the configuration of σ is given in Table II. Unless otherwise specified, the patch size

TABLE II
CONFIGURATION OF σ FOR GENERATING GROUND TRUTH DENSITY MAPS IN EXPERIMENTS

Dataset	Resolution	σ
Bacterial Cell	256×256	3
Embryo Cell	400×400	6
UCSD	238×158	3
UCSD	119×79	1.5
Mall	640×480	12
Mall	320×240	6
Mall	160×120	3
Honeybee	640×480	6
Honeybee	320×240	3
Honeybee	160×120	1.5
Fish	300×410	6
Fish	150×205	3
Seagull	624×964	3

used in all the experiments is set to 6×6 (after PCA, the feature dimension of image patches reduces to 17) and the patch step is set to 3.

Since the kernel function can be viewed as similarity measure, commonly used kernel functions include the linear kernel, polynomial kernels, Gaussian radial basis function (RBF) kernels, and Laplacian kernels which are expressed as follows, respectively: Linear kernel: $k(\boldsymbol{u}, \boldsymbol{u'}) = \boldsymbol{u}^{\mathrm{T}}\boldsymbol{u'}$; Polynomial kernel: $k(\boldsymbol{u}, \boldsymbol{u'}) = (\boldsymbol{u}^{\mathrm{T}}\boldsymbol{u'} + 1)^q$; Radial basis function: $k(\boldsymbol{u}, \boldsymbol{u'}) = \exp\left(-\frac{\|\boldsymbol{u}-\boldsymbol{u'}\|^2}{2\mu^2}\right)$; Laplacian kernel: $k(\boldsymbol{u}, \boldsymbol{u'}) = \exp\left(-\frac{\|\boldsymbol{u}-\boldsymbol{u'}\|}{\mu}\right)$.

Clearly, for the linear kernel, no parameter needs to be set. For the polynomial kernel, the parameter q is set to 2, while the parameter μ for the radical basis function and Laplacian kernels is set as 1.6 and 2.4, respectively.

3) Evaluation Metrics: **Mean absolute error** (MAE) and **mean squared error** (MSE) are commonly used to evaluate the counting performance:

$$MAE = \frac{1}{m}\sum_{i=1}^{m}\|r^i - \hat{r}^i\|_1, \quad MSE = \frac{1}{m}\sum_{i=1}^{m}\|r^i - \hat{r}^i\|_2^2 \quad (20)$$

where r^i is the ground truth counting number of the i_{th} sample and $\hat{r}^i$ is the predicted counting result. m is the total amount of measured samples. Obviously, the lower the MAE and MSE, the higher the counting accuracy.

B. Experiments on Benchmark Datasets

Several experiments are carried out on seven datasets for validating the effectiveness and properties of M-VOC and several mainstream VOC methods, including 1) counting by global regression: RR$^+$ [18]; 2) counting by object density estimation: Dens+MESA* [16], Dens+RF* [19], Codebook+RR* [17], COUNT Forest* [55], Rodriguez *et al.* [56], and Idrees *et al.* [54]; 3) counting by CNN: CNN [27], MCNN [28], CCNN [26], and Hydra 2s [26].

The various versions of M-VOC, such as least square, energy, non-negativity, and sparsity, are denoted as M-VOC(LS), M-VOC(e), M-VOC(nn), and M-VOC(s), respectively.

1) Performance Comparison on the Benchmark Datasets:

a) Bacterial cell dataset and embryo cell dataset: From Table I, it is noted that there are 200 images in the cell dataset. Adhering to the protocol in [16], we choose N (where $N = 1, 2, \ldots, 32$) images randomly from the first 100 images for training, meanwhile the latter 100 images are used for testing. For the remaining data, they are used as the validation set. Experiments are carried out for 5 independent runs. The averaged MAE and MSE are used as the performance metrics. It is noted that the M-VOC only uses the raw data extracted from the blue channel. The experimental results are given in Table III. From this table, we have the following observations. 1) Among the variants of M-VOC, M-VOC(s) outperforms M-VOC(LS), M-VOC(e), and M-VOC(nn). This result suggests that, at least on the cell dataset, sparsity plays a more important role than non-negativity and energy constraints. 2) For overall performance, KM-VOC (RBF) performs better than the other VOC methods. It is noted that when $N = 1$, i.e. one random training image is used, KM-VOC (RBF) achieves minimal counting errors (6.4 ± 1.3). When N is increased, the performance of KM-VOC (RBF) becomes slightly inferior to Dens+RF but superior to or comparable to other methods. When N reaches 32, the result of KM-VOC (RBF) is nearly the same as that of Dens+RF. However, we need to note that KM-VOC (RBF) uses raw data as the features while Dens+RF used the fused features. 3) KM-VOC (RBF) performs much more consistently since they give almost the smallest standard deviation with 5 independent experiments. Moreover, cell images contain strong out-of-focus blur and vignetting [16]. This shows that KM-VOC with RBF is robust against the interference in cell data.

Experiments are also conducted on a real embryo cell dataset from [57], [58]. Since only 11 images are given in this dataset, we use the following two settings in these experiments: choosing randomly 4 or 8 images for training, and the remaining for testing. Same as above, these experiments are conducted for 5 independent runs. Table IV gives the counting results of these methods. It can be observed that our M-VOC methods outperform Density+MESA and Codebook+RR. Specifically, when $N = 4$, M-VOC(RBF) gives the lowest MAE as 12.0 ± 1.8, and when $N = 8$, KM-VOC(Laplacian) gives the lowest MAE as 7.7 ± 5.7.

b) Sparse pedestrian datasets: With the UCSD and Mall datasets, the experimental protocols in [11] are used. Specifically, for the UCSD dataset, frames 601:1400 are used as the training set while the remaining 1200 frames are used for testing. For the Mall dataset, the first 800 frames are employed for training while the remaining frames are used for testing. For M-VOC with the UCSD and Mall datasets, like the above settings, the number of salient patterns l and the regularization parameter λ is set based on the validation data.

The experimental results are given in Table V. It can be observed from this table that, the MAE predicted by KM-VOC(RBF) is 1.48, lower than CNN [27] (1.60) and only second to MCNN [28] (1.07) on UCSD. On the Mall dataset, it is clear that KM-VOC performs the best among

TABLE III
MEAN ABSOLUTE ERRORS (MAE) ON BACTERIAL CELL COUNTING

The superscripts -, + and * are used to indicate the foreground features, fused features (including foreground, textures, etc.), and the local dense features used in the VOC algorithms, respectively. No superscript means only raw data are employed. Same notations are also used subsequently in Tables V and VI.

Method	validation	$N=1$	$N=2$	$N=4$	$N=8$	$N=16$	$N=32$
Dens+MESA* [16]	MESA	9.5 ± 6.1	6.3 ± 1.2	4.9 ± 0.6	4.9 ± 0.7	3.8 ± 0.2	3.5 ± 0.2
Dens+RF* [19]	counting	N/A	$\mathbf{4.8} \pm 1.5$	$\mathbf{3.8} \pm 0.7$	$\mathbf{3.4 \pm 0.1}$	N/A	$\mathbf{3.2 \pm 0.1}$
Codebook+RR* [17]	MESA	9.6 ± 5.9	6.4 ± 0.7	5.5 ± 0.8	4.5 ± 0.6	3.8 ± 0.3	3.5 ± 0.1
M-VOC(LS)	counting	16.4 ± 7.0	11.2 ± 5.4	6.5 ± 3.6	6.1 ± 2.1	5.4 ± 1.5	4.6 ± 0.8
M-VOC(e)	counting	15.8 ± 6.0	11.6 ± 4.2	6.2 ± 2.0	5.1 ± 1.1	4.9 ± 0.5	4.0 ± 0.3
M-VOC(nn)	counting	8.8 ± 3.0	8.0 ± 2.3	6.9 ± 1.2	5.7 ± 1.1	5.3 ± 0.5	4.8 ± 0.3
M-VOC(s)	counting	8.1 ± 3.6	5.9 ± 0.9	4.9 ± 1.1	4.8 ± 0.7	3.9 ± 0.3	3.6 ± 0.1
KM-VOC (poly,$q=2$)	counting	9.8 ± 3.0	8.0 ± 1.9	6.8 ± 2.4	6.0 ± 1.1	5.1 ± 1.8	3.6 ± 0.1
KM-VOC (poly,$q=4$)	counting	11.7 ± 5.0	6.2 ± 1.2	4.4 ± 0.8	4.5 ± 0.3	4.1 ± 0.5	3.9 ± 0.1
KM-VOC (Laplacian)	counting	7.5 ± 2.1	6.5 ± 1.4	6.2 ± 1.6	5.1 ± 1.5	4.0 ± 0.4	4.0 ± 0.3
KM-VOC (RBF)	counting	$\mathbf{6.4 \pm 1.3}$	$4.9 \pm \mathbf{0.7}$	$4.1 \pm \mathbf{0.5}$	3.9 ± 0.4	$\mathbf{3.5 \pm 0.04}$	$3.3 \pm \mathbf{0.1}$

TABLE IV
MEAN ABSOLUTE ERRORS (MAE) ON EMBRYO CELL DATASET

Method	$N=4$	$N=8$
Dens+MESA* [16]	20.2 ± 2.3	17.2 ± 7.7
Codebook+RR* [17]	26.5 ± 3.1	21.2 ± 9.4
M-VOC(LS)	15.0 ± 7.3	13.7 ± 8.3
M-VOC(e)	14.3 ± 4.1	13.2 ± 7.7
M-VOC(nn)	12.3 ± 3.8	10.5 ± 6.2
M-VOC(s)	12.8 ± 3.7	10.7 ± 5.8
KM-VOC(poly,$q=2$)	27.7 ± 4.7	21.0 ± 2.6
KM-VOC(RBF)	$\mathbf{12.0 \pm 1.8}$	$9.4 \pm \mathbf{2.2}$
KM-VOC(Laplacian)	16.4 ± 3.8	$\mathbf{7.7} \pm 5.7$

TABLE V
SPARSE CROWD COUNTING PERFORMANCE COMPARISON

Method	UCSD		Mall	
	MAE	MSE	MAE	MSE
RR$^+$ [18]	2.25	7.82	3.59	19.0
KRR$^+$ [8]	2.16	7.45	3.51	18.1
GPR$^+$ [18]	2.24	7.97	3.72	20.1
CA-RR$^+$ [18]	2.07	6.86	3.43	17.7
IIS-LDL$^+$ [25]	2.08	7.25	2.69	12.1
CNN [27]	1.60	3.31	N/A	N/A
MCNN [28]	**1.07**	**1.35**	N/A	N/A
COUNT Forest* [55]	1.61	4.40	2.50	**10.0**
M-VOC$^-$(s)	2.35	8.40	3.22	16.8
KM-VOC$^-$(poly,$q=2$)	2.49	9.56	**2.49**	**10.0**
KM-VOC$^-$(poly,$q=4$)	2.27	7.92	2.58	11.2
KM-VOC$^-$(RBF)	1.48	3.46	2.70	11.9

all. Again, this result validates the effectiveness of KM-VOC in which positive definite kernels are used. Interestingly, we noted that KM-VOC (RBF) performs better than KM-VOC (polynomial, $q = 2$) on the UCSD dataset while KM-VOC (RBF) performs worse than KM-VOC (polynomial, $q = 2$) on the Mall dataset. These results imply that different kernel functions have different capability in measuring similarity for different image scenes.

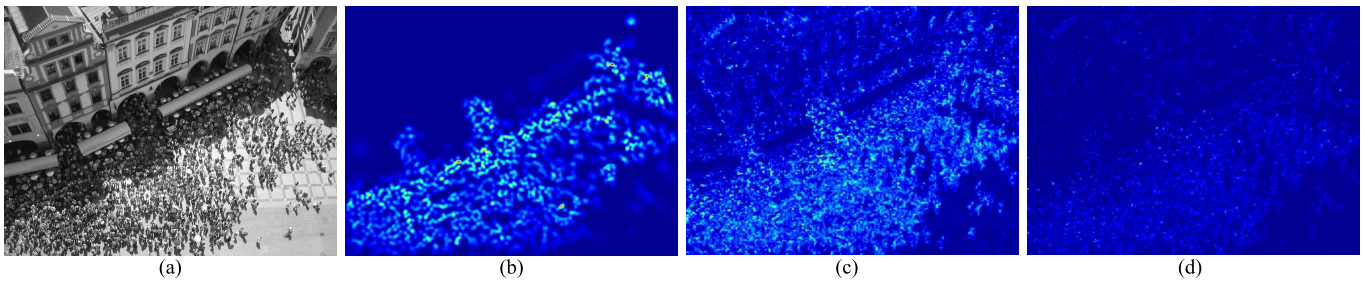
Moreover, following the experimental settings on UCSD in [16], we run another experiment to evaluate our methods. The whole dataset is divided into 4 different training/testing sets: 1) 'maximal': training set consists of frames 600:5:1400; 2) 'downscale': training set is formed by frames 1205:5:1600; 3) 'upscale': training set is composed of frames 805:5:1100; 4) 'minimal': training set is constituted by frames 640:80:1360. The frames outside the training set are used for testing. Experimental results are shown in Table VI. From Table VI, we can see that, compared with the baselines RR, Dens+MESA, and Dens+RF, our KM-VOC(RBF) performs better in max and min (1.65 and 1.80, respectively). KM-VOC(RBF) gets 1.97 in down, better than RR and Dens+RF, and gives 2.24 in up, only lower than RR. Overall, the state-of-the-art methods Codebook+RR, CNN, and COUNT Forest, perform better than KM-VOC(RBF), however, KM-VOC(RBF) gives lower MAE than CNN in max (1.65 versus 1.70).

c) Extremely dense crowd dataset: The UCF_CC_50 dataset contains 50 images depicting crowds in diverse events such as concerts and marathons. As shown

TABLE VI
MEAN ABSOLUTE ERRORS (MAE) ON UCSD DATASET

Method	max	down	up	min
RR$^+$ [18]	1.8	2.34	2.52	4.46
Dens+MESA* [16]	1.7	1.28	**1.59**	2.02
Dens+RF* [19]	1.7	2.16	1.61	2.2
Codebook+RR* [17]	**1.24**	1.31	1.69	**1.49**
CNN [27]	1.70	**1.26**	**1.59**	1.52
COUNT Forest* [55]	1.43	1.30	**1.59**	1.62
M-VOC$^-$(s)	2.01	2.32	2.48	1.82
KM$^-$(poly,$q=2$)	2.19	2.04	2.39	2.48
KM$^-$(RBF)	1.65	1.97	2.24	1.80
KM$^-$(Laplacian)	2.18	2.50	2.21	2.16

in Table I, each image in this dataset has a different number of people ranging from 94 to 4543. In this experiment, we follow the experimental settings used in [54], where 50 images are divided into 5 sets (each of 10 images) randomly, then we performed 5-fold cross-validation on them. The ground truth density map of the given image is computed by the geometry adaptive kernel method proposed in [28] since the object scale varies dramatically in these images, and the geometry adaptive kernel method is able to produce better density maps without perspective information. Experimental results are given in Table VII. It is noted that M-VOCs only achieve benchmark performance and cannot



(a) (b) (c) (d)

Fig. 8. (a) Input dense crowd image. (b) Ground truth density map (count: 1042.57). (c) Estimated density map by KM-VOC (Laplacian) (count: 1639.04). (d) Estimated density map by M-VOC(e) (count: 896.62).

TABLE VII
RESULTS ON UCF_CC_50 DATASET

Method	MAE	MSD	RMSE
Rodriguez et al*. [56]	655.7	697.8	N/A
Dens+MESA* [16]	493.4	487.1	N/A
Zhang et al. [27]	467.0	498.5	N/A
Idrees et al*. [54]	419.56	541.60	N/A
MCNN [28]	377.6	509.1	N/A
CCNN [26]	488.67	646.68	687.59
Hydra 2s [26]	**333.73**	425.26	**283.98**
M-VOC(e)	586.47	94.87	686.85
M-VOC(nn)	672.44	133.51	839.094
M-VOC(s)	649.35	169.89	917.77
KM-VOC(Laplacian)	684.41	115.43	746.68
KM-VOC(RBF)	531.97	**57.71**	645.86

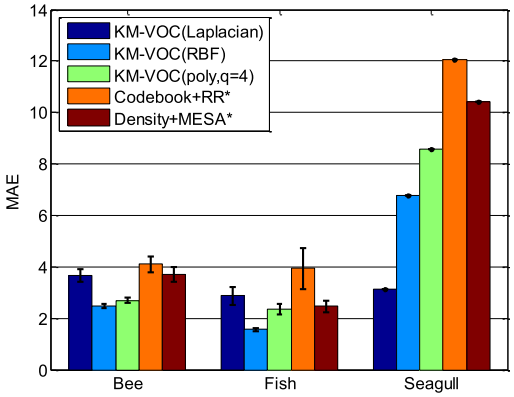


Fig. 9. MAE on Bee, Fish and Seagull datasets.

match the results computed from CNN based methods on MAE and RMSE (the square-Root of the MSE). From our analysis, the images in UCF_CC_50 vary considerably from both appearance and counting number, therefore using simple features is difficult to capture underlying representations of the crowd, as the examples shown in Figure 8. It is notable that KM-VOC(Laplacian) misjudges the building regions and processes them as crowds, as shown in Figure 8(c), while M-VOC(e) gives lower counting result using raw data. From these results, we can see that the manifold assumption used for developing M-VOC may be insufficient for cross scene object counting compared with that for single scene object counting (like UCSD or Mall), which deserves further investigation. In addition, it can be observed that the proposed M-VOC methods give much lower MSD as compared with other baseline methods, even though the MAEs offered by the proposed methods are not the lowest among the compared methods. Specifically, the maximum MSD given by the proposed method is 254.97 which is obtained by the KM-VOC(poly), while the minimum MSD obtained by the baselines is 425.26 which is given by Hydra 2s [26]. The proposed method KM-VOC(RBF) gives a lowest MSD at 57.71, which is approximately one-eighth of that given by Hydra 2s [26] (the lowest among the baseline methods). This indicates that our proposed methods give more stable results as compared with the baselines.

d) Bee, Fish and Seagull dataset: These three datasets are firstly created and applied in [46] for small instances detection. In this study, Dens+MESA and Codebook+RR are taken as baselines. This is because the feature extraction procedure of Dens+MESA and Codebook+RR is standard while global regression based VOC methods need specifically designed features for different counting object types. In addition, they both perform effectively on the Cell dataset when trained with few images (from Table III). For M-VOC, only raw data extracted from the gray channel are used as features. Moreover, λ and l are set via the validation data.

Specifically, the setting for training/testing is given as follows:

Bee: Training on 16 random images chosen from 1:68, and testing on 69:118. The remaining images are used for validation. 5-fold experiments are conducted.

Fish: training on 16 random images chosen from 1:69, and testing on 70:129. The remaining images are used for validation. Similarly, 5-fold experiments are also conducted.

Seagull: training on the first image and testing on the second image. The third image is used for validation.

The experimental results are given in Figure 9. From Figure 9, it is noted that KM-VOC performs better than Dens+MESA and Codebook+RR on counting accuracy for most cases. In addition, the RBF kernel is effective when dealing with different object types.

We also tested our algorithms on the 'Fly' (which is similar to Bee by object types) images [46], but the results, which are similar to those for the above datasets, are not included due to space limitation.

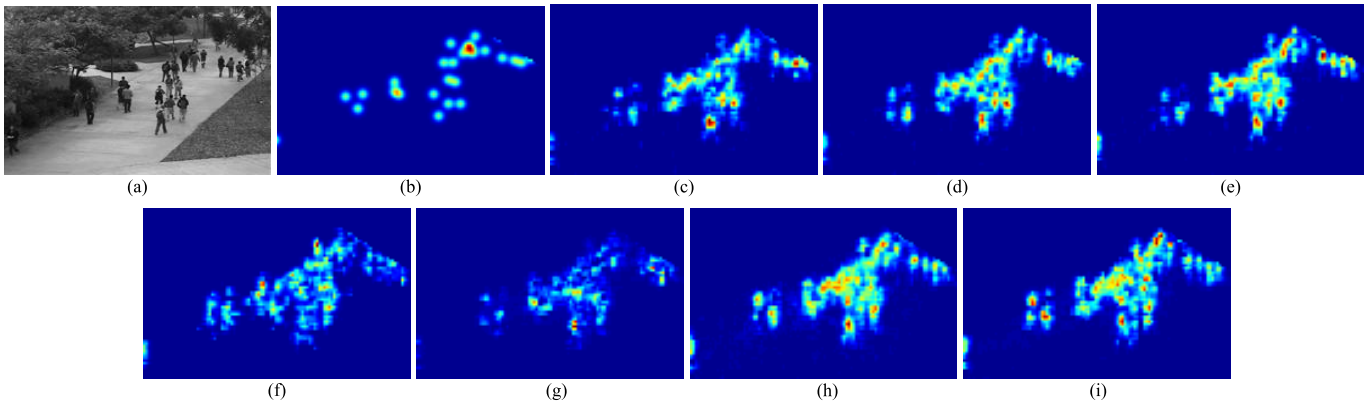


Fig. 10. The estimated density maps of a test image from UCSD. (a) Test image. (b) Ground truth density map (count: 24.79). (c)-(i) are results of M-VOC. They are using LS (count: 25.77), energy (count: 25.47), nonnegativeness (count: 25.08), sparsity (count: 21.99), polynomial kernel (count: 21.96), Laplacian kernel (count: 22.95), RBF kernel (count: 24.83), respectively.

From the experimental results given in Tables III, V, VI, VII, Figure 9, and the discussions above, we are encouraged to see that the M-VOC methods give superior or comparable counting accuracy compared with state-of-the-arts. Moreover, the M-VOC methods still perform reasonably well when the size of training data is reduced. Among all the M-VOC methods, from Table III and Figure 9, we can see that KM-VOC (RBF) yields very accurate counting results even with 1 or 2 training images. In addition, another significant advantage of the M-VOC methods is that they only use simple features, such as the raw data extracted from one color channel or soft foreground features. These results further validate the manifold assumption and local manifold model used in our development of the M-VOC methods, which actually avoids the requirements of engineering different features for different VOC applications. Implicitly, these results validate that the local geometrical similarity between training image patches and their corresponding density maps is an effective and universal prior.

2) Reconstructed Density Maps: To visualize the results of the learned local geometry by using different constraints, several reconstructed density maps are given in Figure 10. It is found the density maps estimated by M-VOC(e) (Figure 10(d)), M-VOC(nn) (Figure 10(e)), KM-VOC(Laplacian) (Figure 10(h)) and KM-VOC(RBF) (Figure 10(i)) look natural while these by M-VOC (LS) (Figure 10(c)), M-VOC(s) (Figure 10(f)) and KM-VOC(poly) (Figure 10(g)) contain artifacts. For example, the density variations and object shapes in Figure 10(f) by M-VOC(s) and in Figure 10(g) by KM-VOC(poly) are discontinuous and unsmooth. Moreover, in Figures 10 and 11, the density maps estimated by KM-VOC(Laplacian) and KM-VOC(RBF) share more similarities with each other than they share with KM-VOC(poly) or M-VOC respectively. This is probably because the RBF kernel and Laplacian kernel are both exponential kernels, while others are not.

3) M-VOC Performance Versus Image Resolution: In many applications, input image resolution varies and we need to evaluate the impact of the image resolution on the performance of M-VOC. It is noted, for our proposed M-VOC methods, we take raw data or foreground feature map as the input feature maps $\boldsymbol{I}^i$, which is expected to be less sensitive to image resolution. To validate this, one experiment for M-VOC(s) is conducted. The experimental results of the MAE versus the image zoom factor are shown in Figure 12. The experimental settings on the Bee dataset are the same as those in Section IV.B.1, and the experimental settings on the UCSD dataset follow the protocols from the training/testing set minimal in Section IV.B.1, while for the Mall dataset, we use the 1:40:800 frames for training, and the 801:12:2000 frames for testing. From Figure 12, it is clear that the MAE results of M-VOC are insensitive to the changes of image resolution in these three datasets. Taking the blue line as an example, when images are downsampled by a factor 4, the MAE result of M-VOC remains almost unchanged.

4) The Impact of Patch Size: It is obvious that patch size and step size are two important parameters for our M-VOC methods. Therefore, in this subsection we evaluate how the patch size affects the performance of the M-VOC methods. For conceptual illustration, without loss of generality, we employ M-VOC(s) on the cell dataset as an example. The experimental setting is the same as that in Section IV.B.1 with $N = 16$, except the variation of the patch size. Specifically, from Figure 13(a), several patch sizes have been tested and it is noted that the MAE keeps nearly steady with the increase of patch size from 4×4 to 7×7, while MAE degrades significantly with a further increase of patch size to 8×8. This is probably because when the dimensionality of feature vector is large enough, the Euclidean distance used in M-VOC would fail to find suitable neighbors since the discrepancy between different vectors can be ignored. Consequently, for our M-VOC methods, a smaller patch size is preferred.

Besides, the step size of patch extraction (step 1 in **Algorithm 1** and **Algorithm 2**) is set to 1/2 patch size (round down to the nearest whole unit) for smoothing the estimated density maps by averaging the overlapping regions.

5) The Impact of the Number of Salient Patterns K: From **Algorithm 2**, we can see that the number of salient patterns K will affect the performance of KM-VOC. In this experiment, to evaluate the impact of K, we vary K from 16 to 1024 with

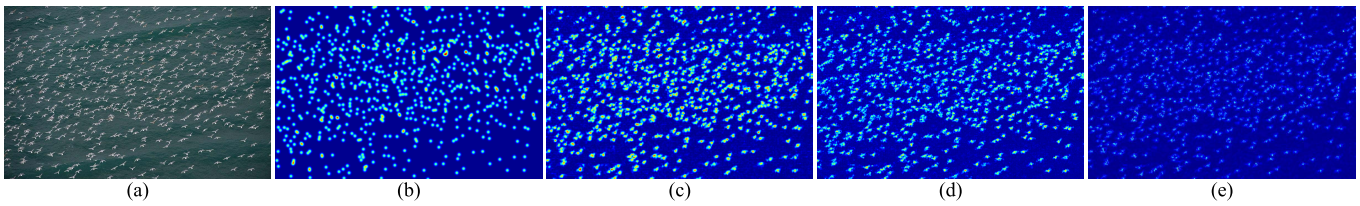


Fig. 11. (a) Input seagull image. (b) Ground truth density map. (c) Estimated density map by KM-VOC (Laplacian). (d) Estimated density map by KM-VOC (RBF). (e) Estimated density map by M-VOC(e).

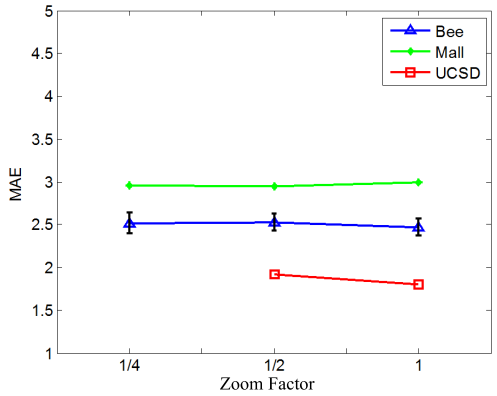


Fig. 12. MAE of KM-VOC (RBF) on the Bee, Mall and UCSD datasets with different image resolution.

Algorithm 2 The KM-VOC Method

Input: Test image $\boldsymbol{X}$, anchored examples set $\{\boldsymbol{C}^t\}_{t=1,2,\ldots,K}$, local examples set $\{\boldsymbol{C}^t\}_{t=1,2,\ldots,K}$, and kernel neighborhood embedding set $\{\boldsymbol{E}^t\}_{t=1,2,\ldots,K}$
Output: Density map $\boldsymbol{X}_d$, the estimated count $c(\boldsymbol{X})$

1: **for** Each input patch $\boldsymbol{x}^i$ (the i_{th} patch) extracted from the test image $\boldsymbol{X}$ **do**
2: Find the neighborhood index t^* for $\boldsymbol{x}^i$ using (19).
3: Compute the density map patch: $\boldsymbol{x}_d^i = \boldsymbol{E}^{t^*} k(\boldsymbol{C}^{t^*}, \boldsymbol{x}_d^i)$. Put $\boldsymbol{x}_d^i$ into $\boldsymbol{X}_d$ at the same location as $\boldsymbol{x}^i$.
4: **end for**
5: Get the estimated density map: $\boldsymbol{X}_d$, and the estimated count: $c(\boldsymbol{X}) = \sum_{z \in \boldsymbol{X}_d} \boldsymbol{X}_d(z)$

a step 64. The experimental settings are as follows: the cell dataset is used; the KM-VOC with RBF is evaluated where μ is set to 1.0, and the number of anchor examples l in each cluster is set to 4096. In addition, λ is set to 1.0 and the patch size is 4×4. The MAE results are given in Figure 13(b), which indicates the trend of the counting accuracy using different K. From Figure 13(b), it is noted that the MAE varies with the changes of K and it reaches the minimum at $K = 576$. We observe that a smaller or larger K will lead to an increase in MAE. Thus, for KM-VOC, K should be carefully selected through cross-validation.

6) The Impact of the Maximal Number of Examples l in Each Neighborhood: In this experiment, we aim to evaluate the impact of the parameter l on the counting performance of our M-VOC methods. Essentially, l is related to the sampling over the subspaces spanned by examples. Undoubtedly, under-sampling would lead to performance degradation as it is contrary to the manifold assumption which requires well-sampling, while over-sampling will degrade counting efficiency as well. As an example, we conducted an experiment on the cell dataset for evaluating the performance of KM-VOC with RBF. The experimental settings are as follows: $\lambda = 1e{-}3$, $K = 256$, $\mu = 1.4$, and $N = 16$. The experimental results are given in Figure 13(c). According to Figure 13(c), we can see that the MAE decreases with the increase of l and when $l > 1024$, MAE becomes stable. However, we also noted that further increasing l does not lead to further decrease in MAE. As a result, in our experiments, l is set as 1024.

TABLE VIII
THE PERFORMANCE (MAE) OF KM-VOC(RBF) ON BACTERIAL CELL, EMBRYO CELL, AND SEAGULL DATASETS WITH DIFFERENT TYPES OF FEATURES

Features	Bacterial	Embryo	Seagull
DSIFT	19.7 ± 2.5	13.3 ± 2.5	43.75
DSIFT+PCA	20.9 ± 3.5	12.8 ± 3.8	43.66
Grad	23.8 ± 5.8	16.7 ± 6.8	6.34
Grad+PCA	21.3 ± 1.6	$\mathbf{9.9 \pm 1.6}$	**5.80**
Raw	$\mathbf{4.9 \pm 0.7}$	12.0 ± 1.8	6.78
Raw+PCA	9.4 ± 4.2	13.0 ± 5.7	6.87

7) Feature Engineering Issues: Here we show how the proposed algorithm performs if different types of features are given as input (rather than raw image patches). To this end, we perform experiments similar to the one shown in Figure 4 for the Bacterial, Embryo Cell, and Seagull datasets. The difference is that the raw image patches are now replaced by the local dense features (dense SIFT), and gradient features (the first and second order derivatives of horizontal and vertical directions). Other detailed experimental settings are the same as those shown in Section III.B.1 for these three datasets, except that the training image number for Bacterial and Embryo Cell datasets is set to 2 and 4, respectively. Table VIII shows the results obtained. It is noted that the KM-VOC method with raw data performs better than that with dense SIFT or gradient based features on the Bacterial Cell dataset, while gradient based features perform better than other two types of features on the Embryo Cell and Seagull datasets. For our proposed M-VOC and KM-VOC methods, it is crucial to ensure that the used features fit with the manifold assumption, in other words, the used features are expected to be able to maintain the local geometry when they are mapped to the density patch domain. Here the similarity between the image patches is decided by the used features when the similarity

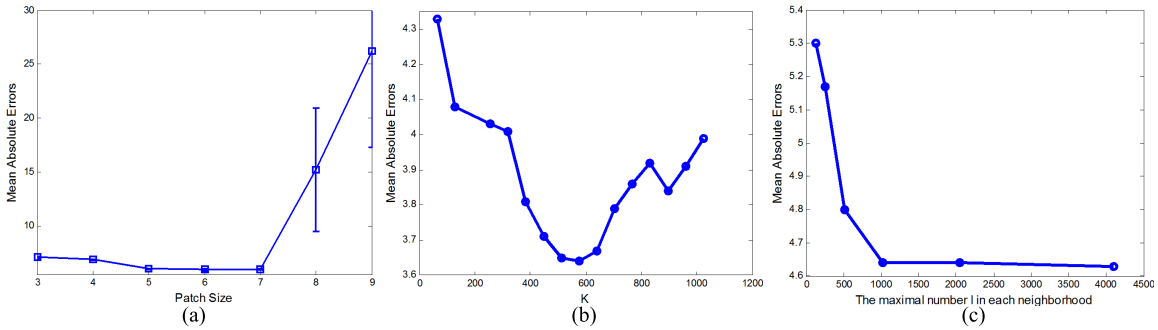


Fig. 13. Some key parameters in M-VOC. (a) The influence of the patch size on counting accuracy. (b) How the number K of salient patterns affects the counting performance on the Cell dataset. (c) The influence of the maximal quantity of examples in each neighborhood on the counting accuracy.

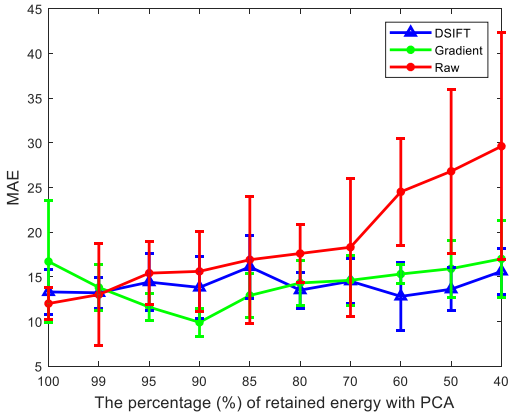


Fig. 14. The MAE versus the percentage of energy retained by PCA using different features for the Embryo Cell dataset.

metric is determined (such as the Euclidean distance). The raw data tend to give fairly good counting results, for maintaining the local geometry of the image patches.

We have also studied the use of PCA to reduce the dimension of the used features and how the number of PCA coefficients affect the counting performance. Apart from its benefit on reducing the computational complexity, PCA is able to alleviate noise and reduce feature redundancy, which might be useful for improving counting results. Similarly, we apply KM-VOC (RBF) with the aforementioned three types of features on the Embryo Cell dataset with the same experimental settings as in Section III.B.1, and the number of training images is set to 4. As shown in Figure 14, for the dense SIFT features, retaining 60% energy gives the lowest MAE 12.8 ± 3.8, while for the gradient features, the lowest MAE 9.9 ± 1.6 is achieved by retaining 90% energy. For the raw data, the lowest MAE 12.0 ± 1.8 is achieved when PCA is not applied. This suggests that the reduction of the dimensionality of the engineered features to a certain degree can help reduce the counting errors. However, over-compressing the dimensionality may lead to ambiguities in nearest neighbor search, and thereby, increased counting errors.

8) Computational Efficiency Evaluation: In this subsection, we show the computational cost of the M-VOC(s) without salient patterns and hierarchical search structure (denoted as M-VOC(e)-nTree), M-VOC(s) with salient patterns and hierarchical search structure (denoted as M-VOC(e)-Tree), and KM-VOC(RBF), using the Cell dataset. The experimental settings of the above methods are identical, and 16 training images are used. The time costs are shown in Table IX, which is an average for 100 test images. All used methods are in MATLAB implementation, and we record the cost of all the methods using the same machine (AMD CPU 4.00 GHz and 16 GB memory).

TABLE IX
COMPUTATIONAL COST ON CELL DATASET

Method	Feature Extraction	Density map reconstruction	Time
Dens+MESA* [16]	9.499 s	0.006 s	9.505 s
M-VOC(e)-nTree	0.084 s	201.242 s	201.326 s
M-VOC(e)-Tree	0.084 s	1.569 s	1.652 s
KM-VOC(RBF)	0.083 s	0.451 s	0.534 s

Table IX shows that KM-VOC(RBF) and M-VOC(e) with hierarchical search structure are two orders of magnitude faster than M-VOC(e) without hierarchical search structure, and one order of magnitude faster than Density+MESA. Specifically, M-VOC(e) and KM-VOC(RBF) take less time than Density+MESA on feature extraction. Using the hierarchical search structure, M-VOC(e) runs much faster than the one without the search structure. In addition, with the precomputed embedding matrices by local regression, the time cost by KM-VOC(RBF) is approximately one-third of that of the M-VOC(s) with the hierarchical search structure.

C. The Properties of M-VOC

According to the above experiments, the proposed M-VOC methods (M-VOC(e), M-VOC(s), M-VOC(nn) and KM-VOC) have the following three desirable properties:

1) They only need a small amount of training data, since the density map of the test image patch is reconstructed over the generalization of a set of examples. In addition, M-VOC performs counting through patches, thus, when the patch size and step size are small, the method still performs well.

2) It is flexible to handle a range of object types including cell, bee, fish, bird, and pedestrian, since only simple features are used, such as raw data or the foreground features, which can be efficiently extracted.

3) The M-VOC methods are robust even for low resolution images and videos. This is because the proposed methods are essentially using the object distribution information obtained

from neighborhood selection and subsequent local geometry representation, which are less affected by variation in image resolutions.

IV. Conclusion

We have presented a manifold-based visual object counting method along with several constraints. The proposed M-VOC method exploits the geometrical prior in images and employs the principle of local embedding to reconstruct the density maps by the local linear representation in the neighborhood. Moreover, to construct more effective neighborhood and overcome the limitations in the local representation for complex background counting problems, nonlinear mapping and kernels are used in M-VOC to reconstruct local geometrical structure in an implicit high dimensional feature space. Extensive experiments on various types of datasets demonstrate that M-VOC is a very promising method for visual object counting.

Acknowledgment

We thank the anonymous reviewers and the associate editor for their helpful comments and suggestions for improving this paper.

References

[1] C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman, "Learning to detect cells using non-overlapping extremal regions," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2012, pp. 348–356.

[2] Z. Lin and L. S. Davis, "Shape-based human detection and segmentation via hierarchical part-template matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 4, pp. 604–618, Apr. 2010.

[3] M. Wang and X. Wang, "Automatic adaptation of a generic pedestrian detector to a specific traffic scene," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 3401–3408.

[4] B. Wu and R. Nevatia, "Detection of multiple, partially occluded humans in a single image by Bayesian combination of edgelet part detectors," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2005, pp. 90–97.

[5] G. J. Brostow and R. Cipolla, "Unsupervised Bayesian detection of independent motion in crowds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2006, pp. 594–601.

[6] V. Rabaud and S. Belongie, "Counting crowded moving objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2006, pp. 705–711.

[7] G. Antonini and J. P. Thiran, "Counting pedestrians in video sequences using trajectory clustering," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 8, pp. 1008–1020, Aug. 2006.

[8] S. An, W. Liu, and S. Venkatesh, "Face recognition using kernel ridge regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2007, pp. 1–7.

[9] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–7.

[10] A. B. Chan and N. Vasconcelos, "Counting people with low-level features and Bayesian regression," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2160–2177, Apr. 2012.

[11] K. Chen, S. Gong, T. Xiang, and C. C. Loy, "Cumulative attribute space for age and crowd density estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 2467–2474.

[12] D. Kong, D. Gray, and H. Tao, "A viewpoint invariant approach for crowd counting," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2006, pp. 1187–1190.

[13] D. Ryan, S. Denman, C. Fookes, and S. Sridharan, "Crowd counting using multiple local features," in *Proc. Digit. Image Comput., Techn. Appl. (DICTA)*, Dec. 2009, pp. 81–88.

[14] D. Ryan, S. Denman, S. Sridharan, and C. Fookes, "An evaluation of crowd counting methods, features and regression models," *Comput. Vis. Image Understand.*, vol. 130, pp. 1–17, Jan. 2015.

[15] C. C. Loy, K. Chen, S. Gong, and T. Xiang, "Crowd counting and profiling: Methodology and evaluation," in *Modeling, Simulation and Visual Analysis of Crowds*. New York, NY, USA: Springer, 2013, pp. 347–382.

[16] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1324–1332.

[17] C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman, "Interactive object counting," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 504–518.

[18] K. Chen, C. C. Loy, S. Gong, and T. Xiang, "Feature mining for localised crowd counting," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2012, pp. 1–11.

[19] L. Fiaschi, U. Koethe, R. Nair, and F. A. Hamprecht, "Learning to count with regression forest and structured labels," in *Proc. IEEE Conf. Pattern Recognit. (ICPR)*, Nov. 2012, pp. 2685–2688.

[20] C. Liu, J. Yuen, and A. Torralba, "SIFT flow: Dense correspondence across scenes and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 978–994, May 2011.

[21] W. Ma, L. Huang, and C. Liu, "Crowd density analysis using co-occurrence texture features," in *Proc. Int. Conf. Comput. Sci. Converg. Inf. Technol. (ICCSCTI)*, Nov./Dec. 2010, pp. 170–175.

[22] Y. Zhou and J. Luo, "A practical method for counting arbitrary target objects in arbitrary scenes," in *Proc. IEEE Conf. Multimedia Expo (ICME)*, Jul. 2013, pp. 1–6.

[23] C. C. Loy, S. Gong, and T. Xiang, "From semi-supervised to transfer counting of crowds," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 2256–2263.

[24] M. Rodriguez, J. Sivic, I. Laptev, and J.-Y. Audibert, "Data-driven crowd analysis in videos," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 1235–1242.

[25] Z. Zhang, M. Wang, and X. Geng, "Crowd counting in public video surveillance by label distribution learning," *Neurocomputing*, vol. 166, pp. 151–163, Oct. 2015.

[26] D. Oñoro-Rubio and R. J. López-Sastre, "Towards perspective-free object counting with deep learning," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 615–629.

[27] C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 833–841.

[28] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 589–597.

[29] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, Nov. 2006.

[30] S. O. Haykin, *Neural Networks and Learning Machines*, vol. 3. Upper Saddle River, NJ, USA: Pearson, 2009.

[31] H. Chang, D.-Y. Yeung, and Y. Xiong, "Super-resolution through neighbor embedding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun./Jul. 2004, p. 1.

[32] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[33] H. Shen, D. Tao, and D. Ma, "Multiview locally linear embedding for effective medical image retrieval," *PLoS ONE*, vol. 8, no. 12, p. e82409, 2013.

[34] S. Conjeti, A. Kazi, N. Navab, and A. Katouzian. (2016). "Cross-modal manifold learning for cross-modal retrieval." [Online]. Available: https://arxiv.org/abs/1612.06098

[35] P. Zhou, L. Du, M. Fan, and Y.-D. Shen, "An LLE based heterogeneous metric learning for cross-media retrieval," in *Proc. SIAM Int. Conf. Data Mining (ICDM)*, 2015, pp. 64–72.

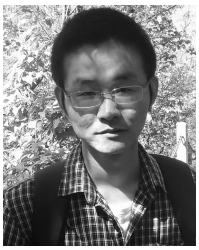
[36] T. Xia, D. Tao, T. Mei, and Y. Zhang, "Multiview spectral embedding," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 6, pp. 1438–1446, Dec. 2010.

[37] Y. Guo, J. Gao, and P. W. H. Kwan, "Regularized kernel local linear embedding on dimensionality reduction for non-vectorial data," in *Proc. Austral. Conf. Artif. Intell.*, 2009, pp. 240–249.

[38] X. Li and L. Shu, "Kernel based nonlinear dimensionality reduction and classification for genomic microarray," *Sensors*, vol. 8, no. 7, pp. 4186–4200, 2008.

[39] K. Q. Weinberger, F. Sha, and L. K. Saul, "Learning a kernel matrix for nonlinear dimensionality reduction," in *Proc. 21st Int. Conf. Mach. Learn.*, 2004, p. 106.

[40] Z. Zhang and J. Wang, "MLLE: Modified locally linear embedding using multiple weights," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2007, pp. 1593–1600.
[41] E. Alpaydin, *Introduction to Machine Learning*. Cambridge, MA, USA: MIT Press, 2014.
[42] R. Timofte, V. De Smet, and L. Van Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 1920–1927.
[43] R. Timofte, V. De Smet, and L. Van Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Proc. Asian Conf. Comput. Vis. (ACCV)*, 2014, pp. 111–126.
[44] Y. Wang and Y. Zou, "Fast visual object counting via example-based density estimation," in *Proc. IEEE Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3653–3657.
[45] Y. Wang, Y. X. Zou, J. Chen, X. Huang, and C. Cai, "Example-based visual object counting with a sparsity constraint," in *Proc. IEEE Conf. Multimedia Expo (ICME)*, Jul. 2016, pp. 1–6.
[46] Z. Ma, L. Yu, and A. B. Chan, "Small instance detection by integer programming on object density maps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3689–3697.
[47] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, no. 9, pp. 509–517, 1975.
[48] R. Timofte, R. Rothe, and L. Van Gool, "Seven ways to improve example-based single image super resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1865–1873.
[49] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2012, pp. 135-1–135-10.
[50] E. Elhamifar and R. Vidal, "Sparse manifold clustering and embedding," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2011, pp. 55–63.
[51] H. V. Nguyen, V. M. Patel, N. M. Nasrabadi, and R. Chellappa, "Sparse embedding: A framework for sparsity promoting dimensionality reduction," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2012, pp. 414–427.
[52] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.
[53] Y. Xu, W. Zuo, and Z. Fan, "Supervised sparse representation method with a heuristic strategy and face recognition experiments," *Neurocomputing*, vol. 79, pp. 125–131, Mar. 2012.
[54] H. Idrees, I. Saleemi, C. Seibert, and M. Shah, "Multi-source multi-scale counting in extremely dense crowd images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 2547–2554.
[55] V.-Q. Pham, T. Kozakaya, O. Yamaguchi, and R. Okada, "COUNT forest: CO-voting uncertain number of targets using random forest for crowd density estimation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3253–3261.
[56] M. Rodriguez, I. Laptev, J. Sivic, and J.-Y. Audibert, "Density-aware person detection and tracking in crowds," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 2423–2430.
[57] E. Bernardis and X. Y. Stella, "Pop out many small structures from a very large microscopic image," *Med. Image Anal.*, vol. 15, no. 5, pp. 690–707, 2011.
[58] C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman, "Learning to detect partially overlapping instances," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 3230–3237.



Yi Wang received the B.E. degree from Northwest A&F University, in 2014, and the M.Sc. degree from Peking University, in 2017. He is currently pursuing the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong. His research interests include computer vision and machine learning.

Yuexian Zou (SM'07) received the M.Sc. degree from the University of Electronic Science and Technology of China, in 1991, and the Ph.D. degree from The University of Hong Kong, in 2000. She is currently a Full Professor with Peking University and the Director of the Advanced Data and Signal Processing Laboratory, Peking University Shenzhen Graduate School. She has been working on several projects related to image processing and machine learning. She has published about 130 academic papers, issued five invention patents, and two of them have been transferred to a company. She conducts several courses for graduate students, such as machine learning and pattern recognition, digital signal processing, and array signal processing. Her research interests mainly in machine learning for signal processing and deep learning and its applications. She was a recipient of the award Leading Figure for Science and Technology by Shenzhen Municipal Government in 2009.

Wenwu Wang (M'02–SM'11) was born in Anhui, China. He received the B.Sc. degree in automatic control in 1997, the M.E. degree in control science and control engineering in 2000, and the Ph.D. degree in navigation guidance and control in 2002, all from Harbin Engineering University, Harbin, China.

He then joined King's College London, London, U.K., in 2002, as a Post-Doctoral Research Associate and transferred to Cardiff University, Cardiff, U.K., in 2004, where he involved in the area of blind signal processing. In 2005, he joined Tao Group Ltd., (now Antix Labs Ltd.), Reading, U.K., as a DSP Engineer, involved in algorithm design and implementation for real-time and embedded audio and visual systems. In 2006, he joined Creative Labs, Ltd., Egham, U.K., as an Research and Development Engineer, involved in 3D spatial audio for mobile devices. Since 2007, he has been with the Centre for Vision Speech and Signal Processing, University of Surrey, Guildford, U.K., where he is currently a Reader in *Signal Processing*, and a Co-Director of the Machine Audition Laboratory. Since 2008, he has been a Visiting Scholar with the Perception and Neurodynamics Laboratory, and the Centre for Cognitive Science, The Ohio State University. He has been a member of the Ministry of Defence, University Defence Research Collaboration in signal processing, since 2009, has been a member of the BBC Audio Research Partnership, since 2011, has been an Associate Member with the Surrey Centre for Cyber Security, since 2014, has been a member of the MRC/EPSRC Microphone Network, since 2015, and has been a member of the BBC Data Science Research Partnership, since 2017.

His current research interests include blind signal processing, sparse signal processing, audio-visual signal processing, machine learning and perception, machine audition (listening), and statistical anomaly detection. He has (co)-authored over 200 publications in these areas, including two books *Machine Audition: Principles, Algorithms, and Systems* (IGI Global, 2010), and *Blind Source Separation: Advances in Theory, Algorithms, and Applications* (Springer, 2014). He is also a Publication Co-Chair of ICASSP 2019 (to be held in Brighton, U.K.). He is currently an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING. He was a Tutorial Speaker on ICASSP 2013, UDRC Summer School from 2014 to 2017, SpaRTan/MacSeNet Spring School 2016, and London Intelligent Sensing Summer School 2017.