Counting pedestrians in crowded scenes with efficient sparse learning

Masamichi Shimosaka, Shinya Masuda, Rui Fukui, Taketoshi Mori and Tomomasa Sato
Department of Mechano-Infomatics,
The University of Tokyo
Email: {simosaka, masuda, fukui, tmori, tsato}@ics.t.u-tokyo.ac.jp

***Abstract*—Counting pedestrians in crowded scenes provides powerful cues for several applications such as traffic, safety, and advertising analysis in urban areas. Recent research progress has shown that direct mapping from image statistics (e.g. area or texture histograms of people regions) to the number of pedestrians, also known as counting by regression, is a promise way of robust pedestrian counting. While leveraging arbitrary image features is encouraged in the counting by regression to improve the accuracy, this leads to risk of over-fitting issue. Furthermore, the most image statistics are sensitive to the way of foreground region segmentation. Hence, careful selection process on both segmentation and feature levels is needed. This paper presents an efficient sparse training method via LARS (Least Angle Regression) to achieve the selection process on both levels, which provides the both sparsity of Lasso and Group Lasso. The experimental results using synthetic and pedestrian counting dataset show that our method provides robust performance with reasonable training cost among the state of the art pedestrian counting methods.**

I. Introduction

In crowded environments, the number of people is a powerful indicator of a crowd, and counting pedestrians is useful for applications, such as environmental improvement, traffic analysis and advertising analysis. Conventional approaches to vision-based monitoring are based on detection [1] or tracking [2] of individuals. However, these approaches are difficult to be used in crowded scenes for several reasons, e.g. the overlap between people, which increase as the increase of people. To resolve this problem, current pedestrian counting techniques in crowded scenes [3] utilize holistic image features, such as the area around people or edges in people regions, and then use regression methods to estimate the number of pedestrians. While arbitrary image features can be leveraged to improve the accuracy thanks to use of regression methods, this leads to the risk of over-fitting issue by unnecessary features. Furthermore, holistic image features are sensitive to the way of foreground region segmentation. Besides, many segmentation methods have some hyper-parameters and the performances are also sensitive to the hyper-parameters. If we use superfluous segmentations to improve the estimation accuracy, it leads to increase run time (segmentation method often take a long time to calculate). In other word, the selection on segmentation leads to fast run time. Hence, careful selection on both segmentations and features from a large number of those is needed. If we can select efficiently the features in a rich dimensional feature space including many relevant features to the count of people, linear regression can be used. For the problem mentioned above, this paper presents an efficient sparse training method via LARS (Least Angle Regression). With our method, we can select useful features and segmentations simultaneously, while determining the weight parameters in the linear regression efficiently.

LARS provides the regularization paths on the solution using sparse regularization terms efficiently. Using sparse regularization terms, e.g. Lasso [6] and variants, are major learning approaches to select useful features while obtaining the linear regression parameters. However, the Lasso methods can not select at both segmentation and feature levels simultaneously. Standard Lasso does not select segmentation. Similarly, Group Lasso [5], which leads to sparse solutions at group level, does not select individual features. Recently, the Sparse Group Lasso [7] that has both sparsity of group and feature is proposed. Nevertheless, sparse Group Lasso is not suitable for an efficient learning due to impossibility of applying LARS method. Sparse Group Lasso has two hyper-parameters and two hyper-parameters search is much complex on the cross-validation. Therefore, we employ a much simpler approach that decouples the problem into two steps: a segmentation selection step and an individual feature selection step. In the first step, our approach reduces segmentation combination patterns from $2^g - 1$ to g by group LARS [5]. In the second step, we obtain the sparse weight parameters in linear regression, which serves as feature selection, by LARS [4]. LARS methods are able to calculate a regularization path efficiently. Thanks to efficient selection at both steps by LARS, our method obtains the solution that has good estimation accuracy within less training time.

In addition, the structure mentioned above is similar to model selection methods. In view of model selection, first, the candidates of useful segmentations are selected. Next, the evaluations on the candidates are calculated respectively. Finally, the best cmbination pattern is determined. This first step corresponds to our segmentation selection step and this second step corresponds to our individual feature selection step by LARS. The simplest combinatorial method is searching solution for all the patterns of segmentations. Indeed, this method can find the best combination, but the search demands an enormous amount of time. The calculation order is $o(2^g)$, where g is the number of input segmentations. Similarly, a forward selection method and a backward elimination method,

as well-known approximated discrete optimization methods, require calculation order of $o(g^2)$[1]. In contrast, the calculation order of our method will be reduced $o(g)$, which is superior to the above approximated methods, while achieving the good estimation accuracy.

II. Two-steps LARS

Our goal is to estimate the number of people, denoted by y. For counting people in crowded scenes, we use linear regression as a model,

$$\hat{y} = \sum_{j=1}^{g} \sum_{i=1}^{d_j} w_{ji}\phi_{ji}(x), \tag{1}$$

where an input vector $\boldsymbol{\phi}_j(x) = [\phi_{j1}(x), \cdots, \phi_{jd_j}(x)]^\mathsf{T}$ indicates a set of features derived from jth segmentation, $\boldsymbol{w}_j(x) = [w_{j1}, \cdots, w_{jd_j}]^\mathsf{T}$ is the weight vector assigned to $\boldsymbol{\phi}_j(x)$. d_j is the number of features extracted from jth segmentation and g is the total number of segmentation methods. $\hat{y}$ is an estimated count and x expresses input image data.

A. Learning overview

The main purpose of this paper is selecting relevant segmentations and features simultaneously and efficiently. To achieve the purpose, it is necessary for learning methods to induce sparse weights at both segmentation and individual feature levels. If w_{ji} is zero in (1), $\phi_{ij}(x)$ does not affect an output $\hat{y}$, which means that the feature $\phi_{ij}(x)$ is useless. In addition, if $\boldsymbol{w}_j$ is a zero vector, all the elements of the assigned input vector $\boldsymbol{\phi}_j(x)$, which indicates a segmentation, are useless. Hence, the sparse solution at two stages, which has many $\boldsymbol{w}_j = \mathbf{0}$ and $w_{ji} = 0$, functions as the selection at both segmentation and individual feature levels. The sparse solution for $\boldsymbol{w} = [\boldsymbol{w}_1^\mathsf{T}, \cdots, \boldsymbol{w}_g^\mathsf{T}]^\mathsf{T}$ functions as a selection of segmentations and sparsity of $\boldsymbol{w}_k = [w_{k1}, \cdots, w_{kd_k}]^\mathsf{T}$ functions as a selection of features.

Given N training data $\boldsymbol{y} = [y_1, \cdots, y_N]^\mathsf{T}$, the weights are generally learned by

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \sum_{i}^{N} (y_i - \hat{y}_i)^2 + \Omega(\boldsymbol{w}), \tag{2}$$

where $\Omega(\boldsymbol{w})$ is the product of a regularization term. Changing $\Omega(\boldsymbol{w})$ leads to a change of weight characteristic. Conventional approach often fixes the $\Omega(\boldsymbol{w})$, but we change $\Omega(\boldsymbol{w})$ at each step, $\Omega_1(\boldsymbol{w})$, $\Omega_2(\boldsymbol{w})$. Then, we obtain the sparse solution at segmentation and feature levels simultaneously using efficient methods for cross-validation, LARS. Our procedures mainly can be divided into two steps. The one is to obtain a sparse solution for promising combinations of segmentations. Another one is a sparse learning for individual features in selected promising segmentations.

In the first stage, our method reduces candidates of segmentation combinations from $2^g - 1$ into g patterns efficiently and let S_k $(k = 1, \cdots, g)$ be a set of selected segmentations. $2^g - 1$ is the number of all the combination patters. In this stage, we utilize $\Omega(\boldsymbol{w}) = \lambda||\boldsymbol{w}||_2^1$ $(\lambda : \infty \to 0)$ and use Group LARS [5] to obtain S_k. Group LARS searches the selected segmentations corresponding hyper-parameter λ changes. Selected groups increase one by one as a hyper-parameter decreases. Then, $|S_k|$ means the number of selected groups, i.e. $|S_k| = |S_{k-1}| + 1$. The total number of changing points which a new group is added at is g points. Group LARS algorithm needs calculations only at the changing points and can obtain the S_k efficiently [5]. The Group LARS does not provide a weights solution to the original problem, but our method uses only the information whether each group is selected or not, which can be obtained by Group LARS.

Algorithm 1 Two-Steps LARS

Input: $\boldsymbol{\Phi}(\boldsymbol{x}) = [\boldsymbol{\phi}(\boldsymbol{x}_1), \cdots, \boldsymbol{\phi}(\boldsymbol{x}_N)]^\mathsf{T}$: features for N frames
$\boldsymbol{y} = [y_1, \cdots, y_N]^\mathsf{T}$: observations for N frame
K : an upper limit number of used segmentations

[Group Selection : Group LARS]
$S_k \leftarrow \mathcal{A}_k$ calculated by Group LARS in [5] $(k = 1, \cdots, K)$

[Feature Selection : LARS]
for all S_k **do**
$\hat{\boldsymbol{w}}_k \leftarrow$sparse learning using cross-validation
$score_k \leftarrow$ MSE scored by cross-validation
end for

Output: $\boldsymbol{w}_{\hat{k}} : \hat{k} = \underset{k}{\operatorname{argmin}}\, score_k$

After acquisition of S_k, our method applies the second step learning to each S_k. In the second step, our method obtains the individual sparse weights for each S_k. Then, our method compares each solution and determines an optimal solution. To obtain the solutions, we use the LARS [4] as an efficient solution method to $\Omega_2(\boldsymbol{w}) = \lambda||\boldsymbol{w}||_1$, which is regularization term for S_k in the second step. LARS method obtains a regularization path corresponding to the change of λ efficiently. Once the regularization path is calculated, the determination of a hyper-parameter with cross-validation is easy and fast. In addition, we also obtain the evaluation for each solution with cross-validation. Finally, we can determine the best optimal solution using the evaluation.

B. Segmentation selection

The segmentation selection step needs an appropriate regularizer to obtain sparsity at a segmentation level, which means $\boldsymbol{w}_j = \mathbf{0}$ for many j. Then, this step uses L1-norm at a segmentation level and

$$\Omega_1(\boldsymbol{w}) = \lambda_1 \sum_{j=1}^{g} \left(\sum_{i=1}^{d_j} {w_{ji}}^2 \right)^{\frac{1}{2}}. \tag{3}$$

We use Group LARS [5] to obtain the selected group by (3) efficiently. In this step, we use only the information whether each segmentation method is selected or not. The selected segmentations depend on a hyper-parameter λ_1. Concretely, as λ_1

[1]Specifically, this method requires $g^2/2$ times Lasso model selection

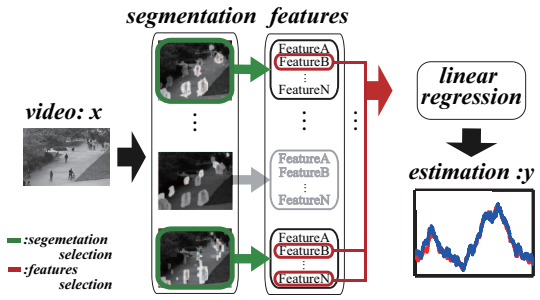

Fig. 1. learning overview

decreases from ∞ to 0, the number of selected segmentations increases one by one [5]. This means that total combination patterns for segmentation are at most g patterns for all λ_1. Group LARS needs calculations at only changing points and enables us to obtain the information of selected segmentations S_k, equivalent to $\mathcal{A}_k$ defined at p.54 in [5], efficiently. S_k is corresponding to Thus, the segmentation selection step reduces candidates of segmentation combination from all $2^g - 1$ to g.

The Group LARS assumes that the feature model matrices in each segmentation are orthonormal. However, model matrices in each segmentation are usually non-orthonormal. Hence, we orthonormalize them with singular value decomposition (SVD) before applying the Group LARS [7]. We can obtain the orthonormal matrix by SVD and obtain the solution $\boldsymbol{w}^* = [\boldsymbol{w}_1^*, \cdots, \boldsymbol{w}_g^*]$ by the Group LARS. Although $\boldsymbol{w}^*$ does not generally provide a solution to the original problem, it is a trivial matter because our method uses only the information whether each group is selected or not.

C. Features selection

In Section II-B, the segmentation combination patterns are restricted to less than g patterns. To obtain sparsity for features within each segmentation method, L1-norm regularization at an individual feature level is used in this step with

$$\Omega_{2,k}(\boldsymbol{w}) = \lambda_{2,k} \sum_{j \in S_k} \sum_{i=1}^{d_j} |w_{ji}|. \tag{4}$$

$\Omega_{2,k}(x)$ has L1-norm at an individual level and yields sparsity for individuals. In the previous step, our method obtains sparse selections of segmentations. Therefore, $\hat{\boldsymbol{w}}_k$ is a sparse solution at both the segmentation and the individual features level.

The solution paths by the Lasso can be computed very efficiently with LARS [4]. The LARS obtains the whole regularization paths efficiently thanks to properties of piecewise linear solution. Once the regularization paths are calculated, it is easy to obtain the optimal solution with cross-validation and the optimal solution is obtained efficiently for each k. In addition, each cross-validation also serves the evaluation of each $\hat{\boldsymbol{w}}_k$. Finally, our method chooses $\hat{\boldsymbol{w}}_{\hat{k}}$ having the best evaluation.

III. Feature extraction

In this paper, features are extracted through segmentations from video. The construction of input vectors consists of three steps. Firstly, the video is segmented into regions moving in different directions. Next, for target crowd segments, various image features are extracted. Finally, we apply a perspective map to weight image according to its approximate size in the real scene.

A. Crowd segmentation

Firstly, we segment the crowds moving in different directions. We adopt eight segmentation methods. Adopted segmentations are the mixture of dynamics textures [8], optical flow and background subtraction. The mixture of dynamics textures is also used in [3] [9] and models a collection of videos consisting of different visual processes. For the mixture of dynamics textures, two segmentations are used from the difference of a target class. Optical flow segmentations are based on the gained velocity vectors. We utilize temporal local optimization method [10], Lucas-Kanade method [11] and Horn-Schunck methods [12], which are used as two segmentations from the difference of hyper-parameters. Segmentation based on background subtraction utilizes contrast between a background image and a target image. In addition, we also use random region segmentation.

B. Image features

Various image features are extracted from segmented regions. We use 25 image features for each segmentation region, such as the amount of segmentation areas, edges in segments and texture information, in this paper. The features extracted are same as those used in [9] and [3].

TABLE I
USED FEATURES

Segmentation region	Internal edge	Texture
area, perimeter, blob, perimeter-are ratio, perimeter histogram (4 ori.)	amount of edges, edges histogram (4 ori.)	homogeneity (4 ori.), energy (4 ori.), entropy (4 ori.)

C. Perspective normalization

The effects of perspective must be considered before we estimate the number of pedestrians by using extracted image features. People closer to the camera appear larger and people farther from the camera appear smaller. To account for perspective, a perspective map is calculated by using the relative sizes of two reference people [3]. The method of making a perspective map utilizes two-dimensional features. For features based on area (e.g. segmentation area), the weights are applied to each pixel. For features based on edges (e.g. edge histogram), the square-roots of the weights are used. By using a perspective map, the estimation method can be applied without the effects of perspective.

IV. Experimental results

A. Evaluation criteria

The performance of the proposed method is accessed by using four criteria: 1) Estimation accuracy, 2) The number of selected segmentation methods, 3) the number of selected features, 4) training time. 1) Accuracy is measured by MSE (Mean Squared Error), borrowing from the previous study [3]. 2) The number of selected groups represents the number of $\boldsymbol{w}_j \neq \mathbf{0}$ in the solutions. The less number of selected groups, the better generalization is obtained and the less computational cost is required. 3) The number of selected features represents the number of $w_{ji} \neq 0$. Less number of selected features is also superior because of a defense against over-fitting. Training time includes determination of hyper-parameters but excludes feature extractions.

B. Comparative methods

In this experiment, the proposed method is compared with other five methods. Ridge regression is a least square optimization with L2-norm regularizer and the derived solution is not sparse. This learning has one hyper-parameter and cross-validation determines it. Group Lasso [5] is one of the sparse learning methods. Indeed the derived solution has grouped sparsity, but the sparsity within each group is unavailable. This learning also has one hyper-parameter and cross-validation determines it. Group forward selection is one of the combinatorial optimization methods for group. Forward selection approach starts with $|S_0| = 0$ and increases the number of used segmentations step by step. To select the new added segmentation, LARS [4] and cross-validation are used at each step. Group backward elimination is similar to Group forward selection. Backward elimination approach starts with $|S_0| = g$ and decreases the number of used segmentations step by step. To select the new eliminated segmentation, LARS [4] and cross-validation are used at each step. In the Group forward selection and the backward elimination, combination patterns, which is calculated with LARS and applied cross-validation, are in proportion to g^2. Gaussian process regression (GPR), a nonparametric kernel based method, is used in the previous work [3]. Since GPR has several kernels and several hyper-parameters related to the kernels, the hyper-parameters are determined by evidence maximization framework. They often have poor optimization result due to the non-convexity. For that, maximizing the likelihood is executed repeatedly with random initialization and the parameters having the best score are chosen in the experiment.

C. Preliminary experiment with synthetic data

We tested our learning method with artificial data to confirm its validity. The artificial data had $N - 300$ training set with $\sum_j d_j = 200$ predictors, in twenty groups having ten features respectively. The coefficients in the half groups were zeros. The number of non-zero coefficients in the other half of groups were (10, 9, 8, 7, 6, 5, 4, 3, 2, 1) respectively, and coefficients and predictors were generated from standard normal distribution. Finally, Gaussian noise with standard deviation 2.0 was added to each observation. MSE (Mean Squared Error) was calculated with 1000 test data. Fig. 2 shows the generated artificial weights and the estimated weights. Table II shows the comparative results. Experimental result shows our proposed method obtains the best accuracy and takes the shortest time of all group selection methods. This is because our method has the efficient LARS selection step, described in Section II-B. In addition, the estimated weights shown in Fig. 2 are sparse at both group and individual levels, which means that our method achieves the selection function at two levels. Hence, our learning method provided robust performance efficiently and quickly while selecting at two levels.

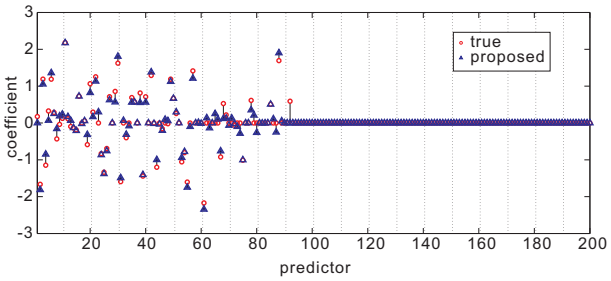

Fig. 2. Generated weights and the estimated weights by our method

TABLE II
PRELIMINARY EXPERIMENTAL RESULT

	MSE	training time (s)
Proposed	5.75 ± 0.38	261± 103
Ridge	10.03 ± 0.66	13 ± 4
G-Lasso	6.94 ± 0.55	539 ± 83
Forward	5.86 ± 0.42	2030±239
Backward	5.84±0.44	2634±355
GPR	9.87 ±0.70	5213 ± 2237

D. Pedestrian Data and Setup

Our test was performed on UCSD pedestrian database [3]. This database contains 2000 frames of pedestrian traffic with crowds of size 11-46 people. The dataset contains the ground truth number of people moving in two directions. The video has been down-sampled to 238×158 pixels and 10fps, gray-scale. A sample frame and a region-of-interest (ROI) are shown in Fig. 3. In this experiment, targets are people moving in away direction from the camera. The training set contains 800 frames, between frame 600 and 1399 with the remaining 1200 frames held out for testing. This training and testing set is the same as Chan [3]. The ground-truth pedestrian counts over time are shown in Fig. 4 with the estimation result.

E. Experimental results and discussion

The result is presented in Table III. In this table, the numbers with bold font represent the first and the second in each section. An example of the estimation result by our method is shown in Fig. 4.

The MSE of the proposed method was the second-best of all the methods. While the GPR-based method achieves the best performance in MSE, the training cost is not negligible.

Fig. 3. Example frame of dataset and Region-Of-Interest (ROI)

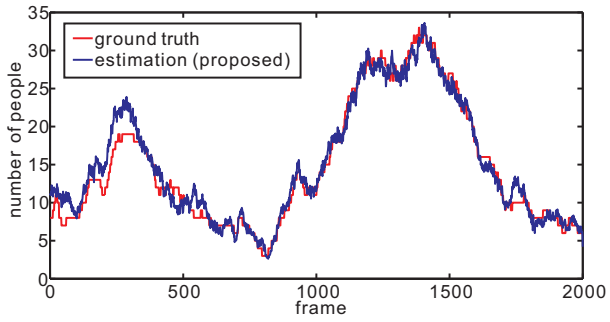

Fig. 4. The ground truth and the estimation result for the test data in the pedestrian dataset

From the point of view in the number about used segmentations, the proposed method reduced the eight segmentation candidates into four segmentations, which is half of the given segmentations. With regard to the used features, the proposed method reduced the number of used features to 6.5%. This shows that the sparsity at both the group and individual levels is achieved (see Fig. 5).

Besides, our approach had the shortest training time except for the Ridge regression. The time was less than one-quarter of that of the second-shortest method having group sparsity thanks to using LARS approach to reduce segmentation selection into $o(g)$. Note that the forward / backward selection methods achieve poorer estimation performance, meanwhile they require more segmentation selection process than ours.

Considering the factors in estimation performance and training cost together, the proposed method is well-balanced in comparison with the state of the art methods.

TABLE III
EXPERIMENTAL RESULT

	MSE	num. of segmentation	num. of features	training time (s)
Proposed	**3.64**	4/8	13/200	**104**
Ridge	4.97	8/8	200/200	**15**
G-Lasso	6.16	4/8	100/200	2068
Forward	4.16	**3/8**	**10/200**	426
Backward	4.11	**2/8**	**9/200**	567
GPR	**3.49**	8/8	200/200	17026

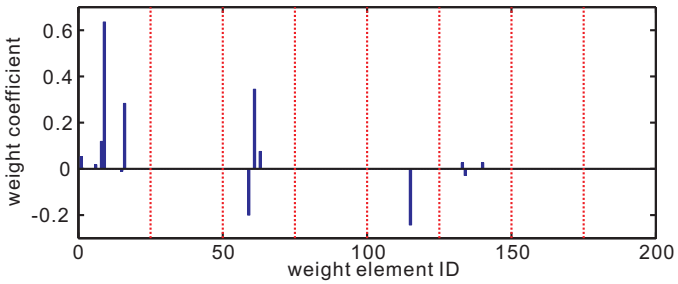

Fig. 5. calculated weights with proposed method: A section between dot lines represents a segmentation. The number of used feature are 200, which is 25 features times 8 segmentations. Selected 4 segmentations, which have a non-zero weight vector, are mixture of dynamics textures, optical flow (temporal local optimization), optical flow HS-1 and optical flow HS-2. HS-1 and HS-2 use different hyper-paremeters. With regard to features, 13 features, for example edge, edge histogram and area, have non-zero weight cofficient. The details about segmentations and features are described in Section III.

V. Conclusion

In this paper, we proposed the two-steps LARS learning method which selects useful segmentations and features with efficiency. The contribution of this paper is a reduction of training time by two-steps LARS while keeping the good estimation accuracy. We applied our method to counting pedestrian using multiple holistic image features in a crowded environment. Besides, we also applied our method to an artificial dataset. The both experimental results show that our method provides robust performance with small number of segmentations and features, while the training cost is reasonable among the state of the art pedestrian counting methods.

References

[1] P. Viola, M. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," in *Proc. ICCV*, vol. 1, 2003, pp. 734–741.

[2] V. Rabaud and S. Belongie, "Counting crowded moving objects," in *Proc. of CVPR*, vol. 1, 2006, pp. 705–711.

[3] A. B. Chan, Z. J. Liang, and N.Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *Proc. of CVPR*, 2008, pp. 1–7.

[4] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Ann. Statist*, vol. 32, pp. 407–499, 2006.

[5] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. R. Statis. Soc. B*, vol. 68, pp. 49–67, 2006.

[6] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. R. Statis. Soc. B*, vol. 58, pp. 267–288, 1996.

[7] J. Friedman, T. Hastie, and R. Tibshirani, "A note on the group lasso and a sparse group lasso," in *Technical report, Department of Statistics, Stanford University*, 2010.

[8] A. B. Chan and N. Vasconcelos, "Modeling, clustering, and segmenting video with mixtures of dynamic textures," *Trans. of PAMI*, vol. 30, no. 5, pp. 909–926, 2008.

[9] A. Chan and N.Vasconcelos, "Bayesian poisson regression for crowd counting," in *Proc. ICCV*, 2010, pp. 545–551.

[10] J. K. Kearney, W. B. Thompson, and D. L. Boley, "Optical flow estimation: an error analysis of gradient-based methods with local optimization," *Trans. of PAMI*, vol. 9, pp. 229–244, 1987.

[11] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. IJCAI*, 1981, pp. 674–679.

[12] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artifical Interlligence*, vol. 17, pp. 185–203, 1981.