2016 International Conference on Control, Automation and Information Sciences (ICCAIS)
October 27-29, 2016, Ansan, Korea

Conditional Marked Point Process-based Crowd Counting in Sparsely and Moderately Crowded Scenes

Yongsang Yoon[‡], Jeonghwan Gwak[†], Jong-In Song[§], Moongu Jeon*
School of Electrical Engineering and Computer Science
Gwangju Institute of Science and Technology
Gwangju 61005, South Korea
{[‡]nil, [§]jisong, *mgjeon@gist.ac.kr}; [†]james.han.gwak@gmail.com

***Abstract*—Crowd density estimation for counting persons, or for determining interactions among persons, groups of people, or crowds has been a challenging problem since persons can be occluded by other persons in (highly) crowded situations. The successful development of such techniques has diverse purposes, such as reassigning limited resources (e.g., public transportation) properly by counting floating population or categorizing the type of events based on the identification of crowd interactions. While existing counting approaches are mostly based on regression models that directly map features to the corresponding class labels, we propose a conditional marked point process (CMPP)-based approach to count individual persons even in moderately crowded scenes. We use a mixture of Bernoulli shape, which is a stochastic model, estimated from the training set with extrinsic shape distribution that determines the size of a shape for the given location in an input image to count the proper number of persons in different types of scenes. The experiment was carried out on PETS2009 which is a well-known public dataset. It was concluded from the experimental results that the proposed approach can be an alternative to the conventional MPP-based approaches.**

Keywords—crowd counting; crowd density estimation; marked point process; crowded scenes

I. INTRODUCTION

Although the problem of counting people in a scene in order to detect and track objects (e.g., persons) has been widely studied, an optimal solution under complex situations (e.g., with inter-person occlusions) has yet to be found. To solve this problem, we devised a method of distinguishing each individual in a group consisting of pedestrians, where the group is one connected-component blob, by combining the foregrounds obtained from a background subtraction method. A lot of existing methods on estimating crowd density are generally based on regression analysis which finds or identifies a relationship between given low-level features and the number of people without considering high-level information. General approaches (e.g., Chan and Vasconcelos [6]) using regression analysis map features of a group into a real number representing the number of people. However, these approaches have a disadvantage in that two blobs having similar size but a different number of people cannot be distinguished using such features. The proposed approach uses a conditional marked point process (CMPP) to consider the body shape and size (in terms of its width and height) which vary depending on the depth of a given location in a scene. The reason for adopting the CMPP is that the aim of this work is to count persons in sparsely or moderately crowded scenes, which can be easily identified by human's perceptual views, rather than extremely crowded scenes. Since the CMPP provides additional information of a person via size modeling, each person in a scene can be found with higher precision. Although the original concept of using the marked point process (MPP) has been introduced in Ge and Collins [1], this work further improves the MPP-based model to improve its performance in counting people.

II. RELLATED WORK

Various approaches to estimating crowds or counting people have been proposed [1–5]. They can be divided into two major groups by their design objective: 1) detecting and counting each person one-by-one [1,4,5], or 2) counting groups or crowds and summing them up [2,3,7]. Liang *et al.* [2] first extracts features using the Speed Up Robust Features (SURF) algorithm [6] from the foreground image and then clusters them. From these clustered features, a regression algorithm gives a model that can tell a numeric estimation of features in a given group. Li *et al.* [3] extracts features (such as edge and texture) from the foreground image. The Gradient-Uniform Local Binary Patterns (G-ULNP), which is a modified version of the original ULNP, with a linear support vector machine (SVM) are used in template matching. The author mentions that the disadvantages of the previous pedestrian detection and matching can be overcome with the proposed features and template matching techniques. In another study [7], various regression model-based methods (e.g., linear regression, Gaussian process regression (GPR), *k*-nearest neighbor and neural networks) and combination of features (e.g., texture, size, edge, keypoint and shape) are compared with each other to determine the optimal technique. Based on an experimental study, they concluded that the GPR-based approach using a combination of four features, excluding the texture feature, could achieve the best accuracy. The simple texture feature was excluded because it is vulnerable to illumination changes. Maddalena *et al.* [4] uses a feet map that gives information on

978-1-5090-0650-2/16/$31.00 ©2016 IEEE

the location and presence of a person. To construct a feet map, a multi-camera environment is required. Multiple scenes observed from multiple cameras are then projected to the ground using geometric information to create a single feet map. In Maddalena *et al.* [4] and Zhao *et al.* [5], a human-shape model is composed of four ellipses consisting of the head, torso and two legs. The width and height of the human shape are determined by a Gaussian distribution. Next, the likelihood is estimated by the number of pixels for each area. However, the human-shape model may be insufficient for expressing a real human shape because the model is simply composed of a set of four ellipses. Our work is based on Ge and Collins [1], which is similar to the work of Zhao *et al.* [5], and uses four body parts. Both employ person-shape models, instead of features such as edges and textures, and adopt a joint probability in estimating their likelihoods. The major difference between the two works is the way of estimating the likelihood. Zhao *et al.* [5] divides pixels into four predefined areas and assigns a different weight to each area. Ge and Collins [1] uses a fixed set of probabilities, and thus, a different probability can be assigned for each pixel. It also uses the Reversible-Jump Markov Chain Monte Carlo (RJMCMC) algorithm [8] and iterates the algorithm at least 500 times or more. Therefore, even if there are no additional persons to be found, the process does not end until the maximum iteration number is reached. To overcome this drawback, this work further devised a method to dynamically control the maximum iteration number according to the scene condition.

III. PROPOSED METHOD

The goal of our proposed method is to cover the foreground pixels in a given image as much as possible with multiple rectangles, while excluding the background pixels in rectangles. Each rectangle is a *configuration* that represents a *person*. This work is primarily based on a MPP for crowd counting, and composed of three steps: 1) estimation of extrinsic parameters, 2) estimation of intrinsic parameters, and 3) inference of a person. Figure 1 shows the overall procedure of the proposed approach. With the given input image (or video), a binary image is produced by first running a background subtraction method, followed by the random selection of a foreground pixel in the binary image to draw a rectangle. The selected pixel is the center of the configuration, and the size of the configuration is estimated using a size model. Next, this configuration is compared to the shape model, and a decision is made as to whether the current configuration should be accepted or rejected. These steps are iterated until the algorithm meets the specified conditions. This method requires two parameters, extrinsic and intrinsic parameters, which provide the size and shape of the body, respectively. An explanation of how to estimate the extrinsic and intrinsic parameters is given in Section 3-*A* and Section 3-*B*, respectively. The process of counting people in a scene using extrinsic and intrinsic parameters is given in Section 3-*C*.

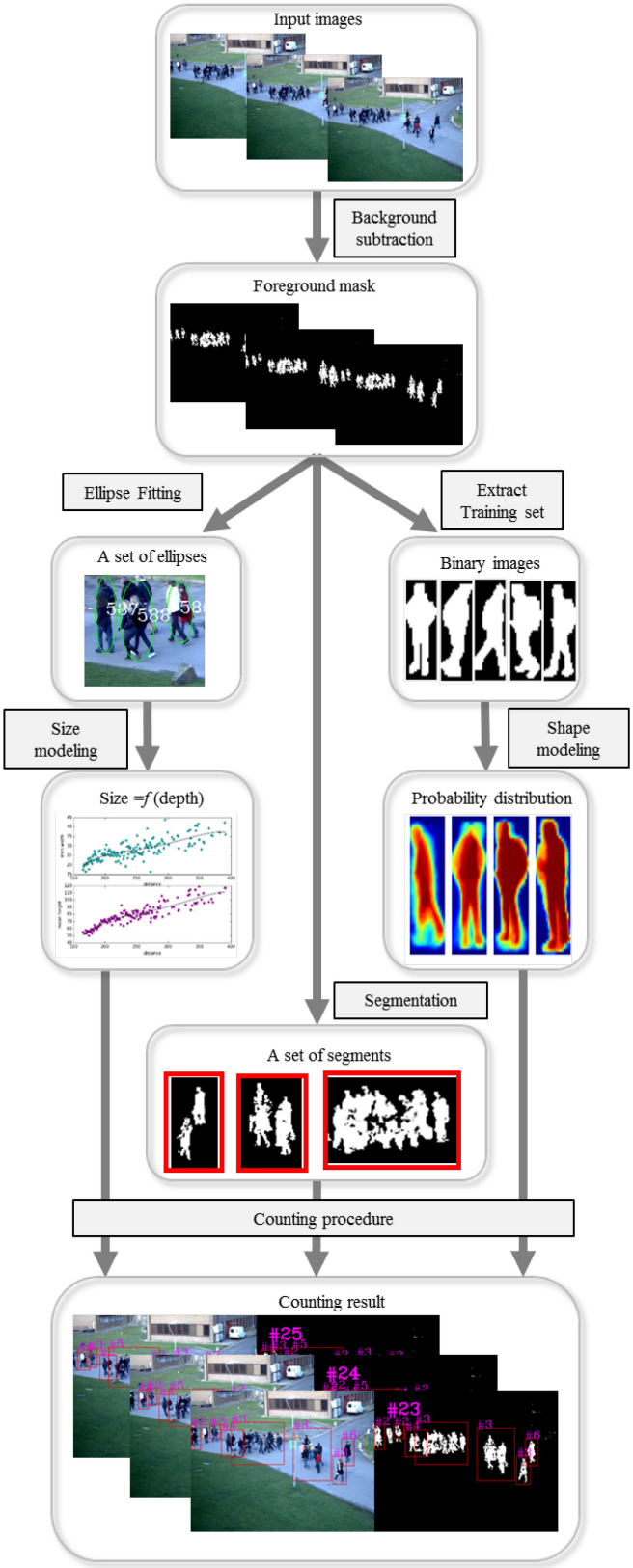


Fig.1. Overall procedure

978-1-5090-0650-2/16/$31.00 ©2016 IEEE

A. Learning Extrinsic Parameter

The extrinsic parameter is a size model, which estimates the width and height of a configuration based on a given foreground pixel, since the size of the configuration varies depending on the depth. In order to compute the appropriate size of the configuration for each pixel, the image depth needs to be computed first, as the object size is inversely proportional to the depth. It is assumed that people are standing on a planar ground, which means that regardless of their moving direction, they are vertical to the ground surface. Based on this assumption, the depth of any given pixel can be simply defined as the y-axis coordinate of the given pixel. First, a foreground mask is produced via background subtraction, and then each blob is fitted to an ellipse. Next, these ellipses are filtered through the constraint of the aspect ratio of a human. Finally, the filtered ellipse set **E** is poly-fitted to obtain the size model, where $\mathbf{E}=\{e_0,e_1,\ldots,e_n\}$ and $e_i=(depth_i,\ size_i)$. The size model is composed of two components -- width model and height model -- as shown in Fig. 2. The colored points and black lines in the graph represent data e_i and the linear model, respectively. The size model takes depth as the input and outputs the size of a person (e.g., width and height).

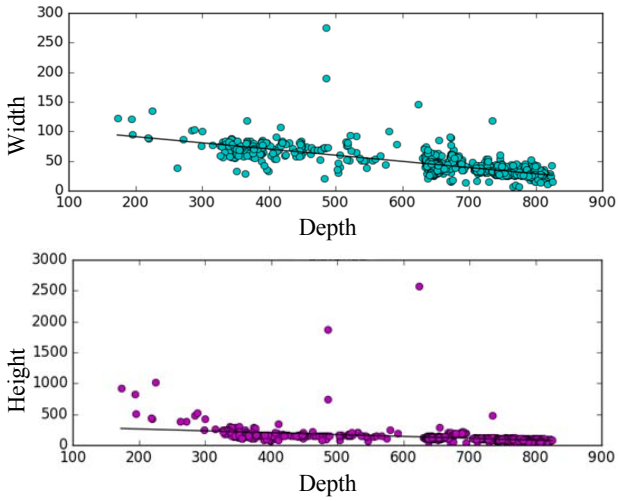


Fig.2. Size model

B. Learning Intrinsic Parameter

While the extrinsic parameter gives size information, the intrinsic parameter provides information on the appearance of a person. One should consider that the boundary pixels of a person have a higher variance than the inner pixels, given that human arms and legs tend to fluctuate when a person is moving. Estimating the intrinsic parameter produces three results: 1) a body shape model, 2) the weight value of each shape model, and 3) a set of variances assisting the shape model, which is obtained from the given training set $\mathbf{X}=\{x_1,\ldots,x_n\}$, where n is the number of training set. The training set **X** is shown in Fig. 3-(a). Each training data x_i consists of binary values, one for the foreground pixel and zero for the background pixel. $x_i=\{a_1,\ldots,a_d\}\in R^d$, where $a_j\in\{0,1\}$. The white and dark areas in the training data x_i in Fig.3-(a) represent the foreground and the background, respectively.

For the body shape model, the Bernoulli mixture model from Ge and Collins [1] was used. Similar to the Gaussian mixture model, the Bernoulli mixture model consists of multiple Bernoulli models. In each Bernoulli model $\mu_i=\{\mu_{i1},\ldots,\mu_{iD}\}$, D is the dimension of the shape model, which is a set of probabilities to be a foreground for each pixel, and i=1,..,k. The set of variances $\mathbf{v}=\{v_1,\ldots,v_D\}$ consists of variances of binary values for the given training set. The weight set of the mixture model is $\boldsymbol{\pi}=\{\pi_1,\ldots,\pi_k\}$. The likelihood function is defined as (1).

$$p(x\mid\mu)=\prod_{d=1}^{D}\mu_d^{x_d v_d}\left(1-\mu_d\right)^{(1-x_d)v_d}. \tag{1}$$

Based on (1), the expectation log-likelihood function in this work is defined as follows:

$$E_z\left[\log p(X,Z\mid\mu,\pi)\right]=\sum_{n=1}^{N}\sum_{k=1}^{K}Z_{nk}\left\{\log\pi_k+\sum_{d=1}^{D}v_d x_{nd}\log\mu_{kd}+v_d(1-x_{nd})\log((1-\mu_{kd})\right\}. \tag{2}$$

E-step: Update latent variable **Z**.

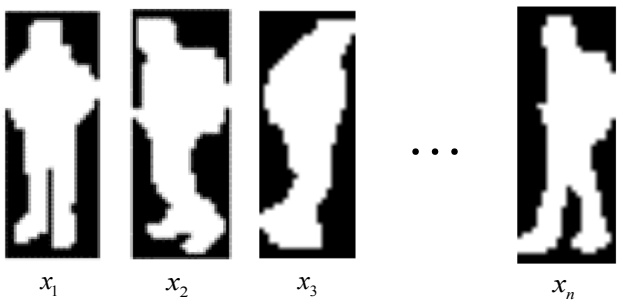
$$Z_{nk}=\frac{\pi_k\times p(x_n\mid\mu_j)}{\sum_{j=1}^{K}\pi_j\times p(x_n\mid\mu_j)}. \tag{3}$$

M-step: Update the parameters to maximize expectation

$$u_{kd}=\frac{\sum_n Z_{nk}x_{nd}}{\sum_n Z_{nk}}. \tag{4}$$

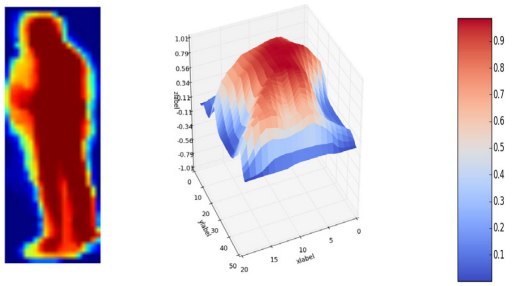
$$\pi_k=\frac{\sum_n Z_{nk}-1}{N-K}. \tag{5}$$

To learn an intrinsic shape model, we used the EM algorithm by iterating both *E-step* and *M-step* to maximize (2). The intrinsic shape model is shown in Fig. 3-(b) in a color map. The indicator in Fig. 3-(b) shows that if the color of a pixel approaches red, it has a higher probability than a pixel which is relatively closer to blue.



(a)

978-1-5090-0650-2/16/$31.00 ©2016 IEEE



2D distribution 3D distribution Indicator

(b)

Fig.3.(a) training data **X** , and (b) shape model.

C. *Inference*

To count a human object, a human appearance is required. The human appearance can be decomposed into two main parts, size and shape, which serve as extrinsic and intrinsic parameters, respectively.

1) Separation. First, the foreground mask, which is a binary image (i.e., foreground and background represents 0 and 1, respectively), is produced by background subtraction. It may contain multiple connected-component blobs, but the assumption is made that each blob is independent of each other. Based on this assumption, they can be separated into independent single blobs to reduce the computation complexity, since each blob can be considered separately, instead of considering all of the blobs at once. Next, a rectangular-shape area called segmentation is confined, and a given connected-component blob without any margin is surrounded. In other words, it is assumed that no people exist outside of this segmentation. Segmentation with no margin, however, may cause a problem. Although a new configuration could describe a person quite well, it would be rejected if it contains a single pixel from outside the segmentation, given the assumption that people only exist within the segmentation. To avoid this undesirable rejection, the size of the segmentation is extended proportionally to its original size when the segmentation is created.

2) Creation of configuration. In the segmentation area, one foreground pixel is randomly selected, followed by the extraction of a specific area, where a person could be found, based on the selected pixel. The size of this area is determined by the size model, which gives the size corresponding to the given location in the image (i.e., the depth, where size varies depending on the depth). Once a pixel has been selected, it and its neighboring pixels are marked as 'visited' to avoid a revisit, thus, reducing the computation complexity and cost.

3) Determination of configuration. Before comparing the created configuration to the shape model, the configuration is checked to see if it contains pixels from outside the segmentation. If the configuration x does not contain any external pixels, it is then compared to the shape model y, which is selected randomly from the shape model set **Y**. The configuration consists of background and foreground pixels, since it is extracted from the given binary image produced from background subtraction. However, since the shape model **Y** consists of probabilities, as described in Section 3.2, **Y** needs to be converted to a binary shape model. Every pixel x_i in the configuration x has its own counterpart pixel y_i in the shape model y. If both x_i and y_i are identical as either the background or the foreground, it is called a *hit*-pixel; if not, it is called a *miss*-pixel, as described in (6).

$$f(x_i, y_i) = \begin{cases} hit & if\ x_i = y_i \\ miss & otherwise \end{cases}. \tag{6}$$

Then, the numbers of *hit*-, *miss*-, *duplicated*-pixels, are counted, respectively. The decision to accept, reject, and update a configuration is made by score S.

$$S = \frac{\omega_h \times \Delta(hit) + \omega_m \times \Delta(miss) - \omega_{dup} \times \Delta(d)}{|\Delta(hit)| + |\Delta(miss)|}, \tag{7}$$

$$S = \frac{\omega_h \times n(hit) + \omega_m \times (D - n(hit)) - \omega_{dup} \times n(d)}{D}, \tag{8}$$

$$S = \frac{\omega_h \times \Delta(hit) + \omega_m \times \Delta(hit) - \omega_{dup} \times \Delta(d)}{D}, \tag{9}$$

where, ω_h , ω_m and ω_{dup} are weight values for the *hit*-pixel, *miss*-pixel and *duplicated*-pixel, respectively; D is the total number of pixels in a frame; $n(hit)$ is the number of *hit*-pixels in the previous configuration set, including the newly proposed configuration; and $\Delta(hit)$ is the number of differences between the *hit*-pixels of the previous configuration set, including the newly proposed configuration, and the previous configuration set, while $\Delta(miss)$ is the difference of *miss*-pixels and $\Delta(d)$ is the difference of *duplicated*-pixels. Since $\Delta(\cdot)$ could have a negative value, the absolute value is used when normalizing the score. A few configurations could be overlapping other configurations. To avoid a significant overlapping problem, we assigned a penalty on the *duplicated*-pixels (d), which indicates the ones found in the overlapping area, to control the score. To have more *hit*-pixels and less d-pixels, our proposed method creates a new configuration, or removes an existing configuration. The above procedure (from Sections 3-*C-1* to 3-*C-3*) iterates until 80% of the foreground pixels in a given segmentation are visited, or a maximum iteration is reached. We intend to construct a robust algorithm that can count people accurately in a crowded situation containing multiple occluded objects.

IV. Experimental Results And Discussion

To verify the effectiveness of the proposed method for crowd counting in sparsely and moderately crowded scenes, the experimental results were tested on the PETS2009 dataset. The results are shown in Fig. 4 and include 6 experimental sets where each set has a different weight vector **V**=[ω_h , ω_m , ω_{dup}].

978-1-5090-0650-2/16/$31.00 ©2016 IEEE

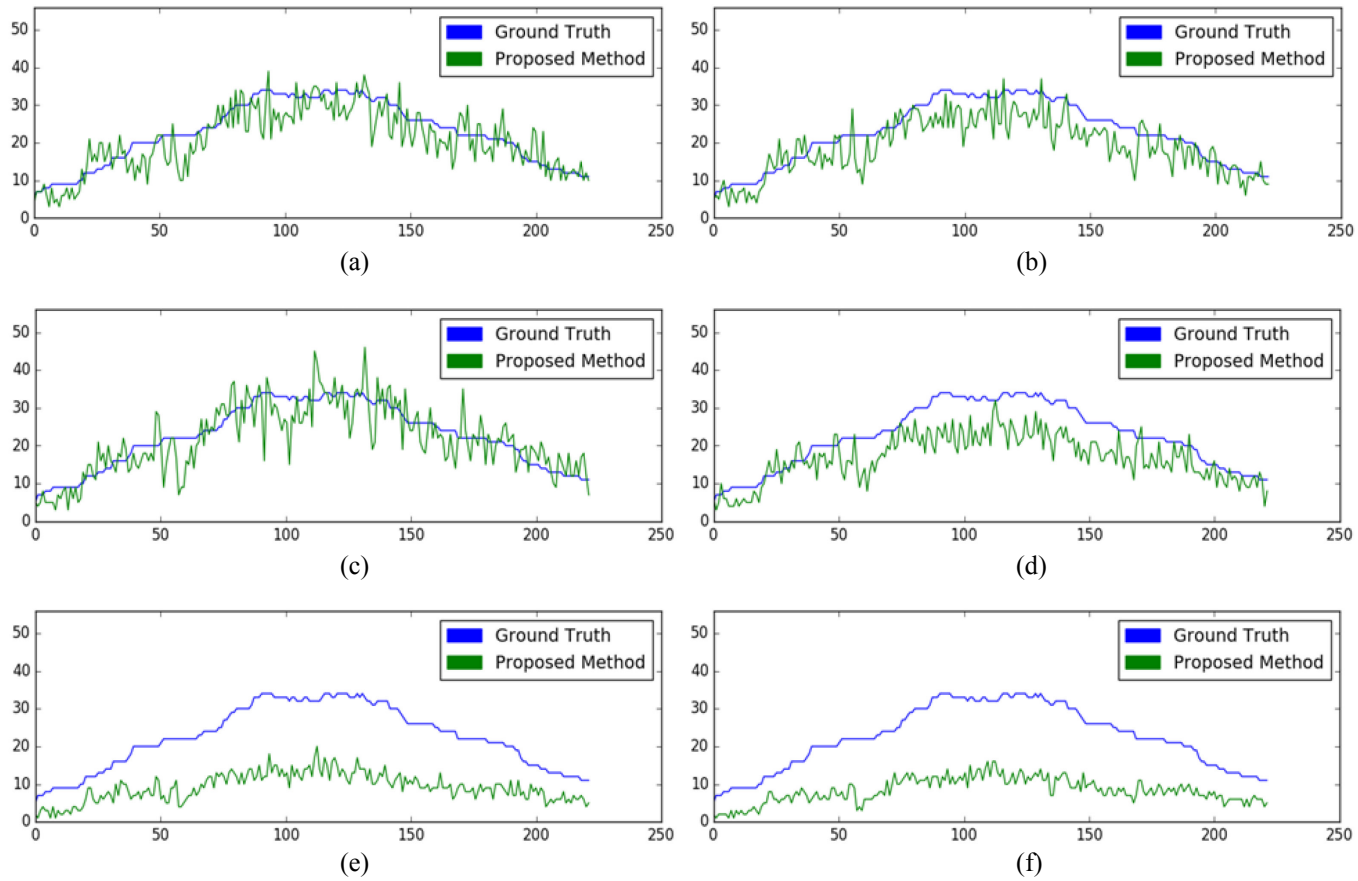


Fig. 4. Simulation results of the proposed method using different score formulas and weight vectors.

Fig. 4-(a) to 4-(d) show the results of using (7) with **V**=[0.95, 0.5, 0.2], [0.8, 0.2, 0.2], [0.99, 0.01, 0.2] and [0.9, 0.1, 0.4], respectively. Figure 4-(e) is the result of (8) with **V** = [0.95, 0.05, 0.2], and Fig. 4-(f) is the result of (9) with **V** = [0.9, 0.1, 0.3]. As we can see from Fig. 4, the proposed method can serve as an alternative method to the conventional approach [1] and reduce the computation load.

From several simulations, we found that varying the value of ω_h has a negligible effect on the counting results. Therefore, ω_h does not have significant impact once a threshold (0.8 in our experiments) is reached. However, varying the value ω_{dup} could have a big impact on the results, because accepting a newly created configuration will increase both the *hit*-pixel(s) and the *duplicated*-pixel(s). In particular, the effect of *duplicated*-pixels was more significant due to their severe occlusion in highly crowded situations, reducing the score of the current configuration set and making it difficult to count people in crowded scenes.

The difference between (7) and (9) is attributed to their normalization terms and that between (8) and (9) is attributed to their pixel terms. The drawback of using (8) -- which counts the number of *hit*-, *miss*-, and *duplicated*-pixel -- is the normalization term. The method using (8) tends to accept many new configurations as possible even though the configuration has more *miss*- and *duplicated*-pixels than *hit*-pixels because of the constant normalization value. It is difficult to see a configuration as a person if the *miss*-pixels make up more than half of the configuration size. However, the method will accept the configuration because ω_h is much greater than ω_m or ω_{dup}. As a result, the previous step receives a higher score than the current step, making it difficult for a new configuration to be accepted. This is the main reason why the numerical value of the crowd counting method is lower than the ground truth (GT), as shown in Fig.4-(e). To avoid this drawback, $\Delta(\cdot)$ is introduced in (8). It is better to consider pixel changes only between the previous and the current configuration sets, rather than just counting the overall pixels, such as *hit*-pixels. However, the normalization issue is still present, as shown in Fig.4-(f). Thus, we adopted $\Delta(\cdot)$ only in (7) to estimate the score and its normalization term.

V. Conclusions and Future Work

This work attempts to devise an algorithm for crowd counting in sparsely and moderately crowded scenes using CMPP with modified score functions, and its effectiveness was demonstrated from the experimental results. The proposed method assumes that each segmentation in a frame, as well as each frame, is independent. In other words, no prior information on the previous frame is provided in estimating the current frame. As the results showed that this can lead to inconsistencies in crowd counting estimations, our future work

978-1-5090-0650-2/16/$31.00 ©2016 IEEE

will attempt to utilize previous frames in the estimation (e.g., by modeling and using motion information). Finally, our current research is being carried out to further improve the accuracy and robustness of crowd counting by incorporating evolutionary approaches. Specifically, the birth, death, and update procedures, which were originally proposed in [1] and adopted in this work, are being designed by using evolutionary algorithms.

ACKNOWLEDGMENT

This work was supported by the ICT R&D Program of MSIP/IITP (Grant No. B0101-15-0525, Development of global multi-target tracking and event prediction techniques based on real-time large-scale video analysis) and the Brain Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2016M3C7A1905477), and the Institute of Integrated Technology (IIT) Research Project through a grant provided by GIST in 2016.

REFERENCES

[1] W. Ge and R.T. Collins, "Marked point processes for crowd counting," in Proceeding IEEE international confereince on Computer Vision and Pattern Recognition, 2009, pp. 2913–2920.

[2] R. Liang, Y. Zhu, H. Wang, "Counting crowd flow based on feature points", Neurocomputing, 2014, vol. 133, pp. 377–384..

[3] J. Li, L. Huang, and C. Liu. "Robust people counting in video surveillance: Dataset and system," IEEE International Conference on Advanced Video and Signal-Based Surveillance, 2011 pp.54–59.

[4] L. Maddalena, A. Petrosino and F. Russo, "People counting by learning their appearance in a multi-view camera environment", Patter Recognition Letters, 2014, vol. 36, pp. 125–134.

[5] T. Zhao and R. Nevatia. "Bayesian human segmentation in crowded situations," IEEE conference on Computer Vision and Pattern Recognition (CVPR), Jun. 18-20, 2003.

[6] D. Ryon, S. Denman, S. Sridharan, C. Fookes, "An evaluation of crowd counting methods, featues and regression models," Computer Vision and Image Understanding, 2015, vol. 130, pp. 1–17

[7] AB. Chan and N. Vasconcelos, "Counting people with low-level features and Bayesian regression," IEEE Transactions on Image Processing 21.4 (2012): 2160–2177.

[8] PJ. Green, "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination" Biometrika, 1995, vol. 4, pp. 771–732.

[9] H. Bay, T. Tuytelaars, and L.V. Gool, "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding, 2008, vol. 110, pp. 346–359