2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance

Water Filling: Unsupervised People Counting via Vertical Kinect Sensor

Xucong Zhang[1] Junjie Yan[1] Shikun Feng[1] Zhen Lei[1,2] Dong Yi[1,2] Stan Z. Li[1,2*]
[1]CBSR & NLPR, Institute of Automation, Chinese Academy of Sciences
[2]China Research and Development Center for Internet of Thing
{xucong.zhang1990}@gmail.com, {jjyan,skfeng,zlei,dyi,szli}@cbsr.ia.ac.cn

Abstract

People counting is one of the key components in video surveillance applications, however, due to occlusion, illumination, color and texture variation, the problem is far from being solved. Different from traditional visible camera based systems, we construct a novel system that uses vertical Kinect sensor for people counting, where the depth information is used to remove the affect of the appearance variation. Since the head is always closer to the Kinect sensor than other parts of the body, people counting task equals to find the suitable local minimum regions. According to the particularity of the depth map, we propose a novel unsupervised water filling method that can find these regions with the property of robustness, locality and scale-invariance. Experimental comparisons with mean shift and random forest on two databases validate the superiority of our water filling algorithm in people counting.

1. Introduction

People counting has been a key component in video surveillance applications such as people flow monitoring and tourists flow estimation. Previous methods of people counting based on visible light images or videos can be grouped as following: counting by detection, counting by regression and unsupervised tracker.

The most convenient approach is counting by detection, where multi-scale windows slide over the whole image and a binary classifier is adopted to determine whether there is a people within the window [10, 21]. Once we have detection results, the counting problem is solved naturally. However, there are two unsolved problems in object detection. Firstly, one object may correspond to different bounding boxes and currently the commonly used NMS(non-maximum suppression)[13, 11] tends to become invalid in complex situation where multiple people have interactions with each other. Secondly, some objects may be partially or even wholly occluded so that they are too difficult to be detected. Both of the two problems affect the counting performance.

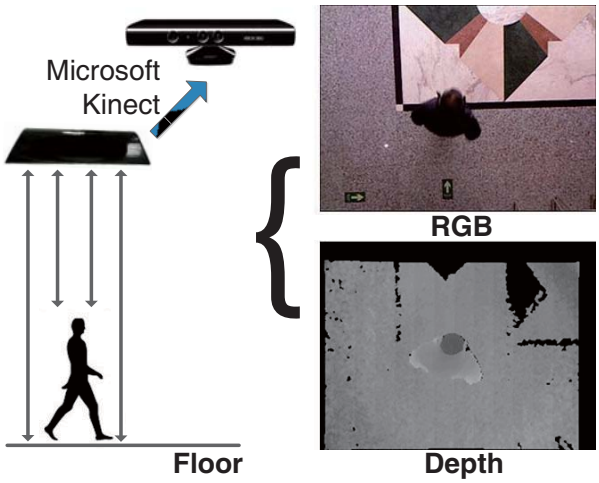


Figure 1. Overview of our people counting system. We place the Kinect on the ceiling and pointing to floor. Right-up and Right-down image are RGB and depth image generated by Kinect sensor respectively. In our system, depth information is used to achieve appearance invariant.

Instead of solving the difficult detection problem, counting by regression methods solves the counting problem directly. Most of the algorithms use regression techniques to learn a map between features and the number of people in the training set and then use the map to estimate the number of people in novel test images or videos [7, 14, 16, 12]. In [5], Gaussian process is used to learn the map between holistic features and the number of people, and in [6] a Bayesian Gaussian process is further proposed to improve the performance. A discriminative training framework is proposed in [15] and the counting people equals to integrate over the foreground region. These methods take the counting problem as a black box. Although many methods have been proposed to increase the generalization capability, the performance always depends on the training instances, and labeling a lot of training instances is time consuming.

*Stan Z. Li is the corresponding author

978-0-7695-4797-8/12 $26.00 © 2012 IEEE
DOI 10.1109/AVSS.2012.82

To avoid reliance on the training instances, some unsupervised methods have been proposed. These methods use point tracker to track visual features and then cluster the trajectories. Under the independent assumption, the number of cluster centers is taken as the number of people. In [17], KLT tracker and agglomerative clustering were used. An unsupervised Bayesian approach was further proposed in [4]. However, the basic assumption does not always hold since that one people may have large actions that results in two clustered trajectories and some people may passed together that results in a single trajectory.

The three approaches can achieve some success in specific situations. However, due to the occlusion, variation of illumination, color and texture, the problem is far from being solved in various applications.

Refs. [2, 20] reported the use of vertical Time-of-Flight depth sensor to detect and track people. Our Kinect sensor is cheaper and high-precision, and the algorithms they proposed are different with ours.

In this paper, we adopt vertical camera since occlusion problem is naturally solved in the view of vertical camera [1]. On the other hand, we apply the Kinect sensor for people counting. By using the structural light technique, Kinect sensor [9] gives the calibrated depth information of every pixel and this information is illumination, color and texture invariant. The sensor was firstly used in [18] and achieved surprising result on human pose estimation. Since the heads are always closer to the camera than other part of the body in the view of vertical Kinect sensor, detecting people's head equals to finding the suitable local minimum regions in the depth image. The system is illustrated in Figure 1.

The vertical depth information generated by Kinect sensor can simplify the people counting problem, but there are still problems in real application. This is because people in the same scene may have various scales or depth information, and the crowded people will make a complex depth map with multiple local extremum. Besides, the raw 3D data from Kinect sensor have a lot of noises, which makes the depth map to be discontinuous. So it is difficult to achieve good performance with the traditional clustering method such as mean shift or watershed segmentation, which is sensitive to the threshold and easily falls into the local optimum caused by the noise. In this paper, we propose a novel algorithm that can effectively find local minimum regions with the advantage of locality, scale-invariance and robustness. Our algorithm is motivated by the water filling process, that the water moves away from the heave and out to the nearby hollow under the force of gravity until the gravitational potential energy can't be reduced any more. We simulate the rain by generating the raindrop according to a uniform distribution. Once a raindrop arrives, we compare its landing spot with its neighborhood and find the descent direction until it can't descend any more, then the the number of raindrop at the balance spot increases. Since most of the raindrop at nearby landing spot tend to flow to the same hollow, we further propose a fast algorithm that speed up the process.

The rest of the paper is organized as follows: Section 2 gives an overview of our system and the people detection model corresponding to the system. Our water filling inference algorithm is discussed in Section 3. Experiments are given in Section 4 and finally in Section 5, we give our conclusion.

2. Problem Formulation

We take the depth image as a function f, where $f(x, y)$ stands for the depth information of pixel (x, y). Due to the noise of Kinect sensor, $f(x, y)$ can be non-derivable or even discontinuous. Finding people in depth image equals to finding local minimum regions in f. Mathematically, the problem can be defined as finding the region A and N that satisfy the following constraint:

$$E_A(f(x, y)) + \eta \leq E_{N \setminus A}(f(x, y)) \tag{1}$$

where $A \in N$, A is the local region and N is its neighborhood. $E(\cdot)$ is an operation to pool the depth information in the region to a real value that reflects the total depth information in the region. η is a pre-defined threshold to ensure that depth in A should lower than $N \setminus A$ with a margin.

Note that A and N can be of arbitrary shape, and finding all the regions in image can be very time consuming. In the following, we will give our water filling algorithm that can effectively find all the suitable regions.

3. Inference via Water Filling

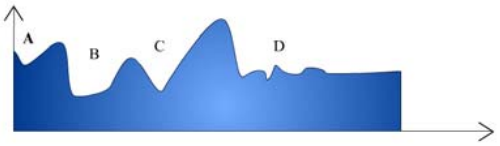
In order to solve the problem effectively and be robust to noise, we introduce an additional measure function $g(x, y)$ to "measure" $f(x, y)$. Mathematically, $g(x, y)$ is defined as:

Definition 1 *$g(x, y)$ is a measure function of $f(x, y)$ if and only if $\exists \varepsilon > 0$, $\forall (x_1, y_1), (x_2, x_2)$, s.t. $\|(x_1 - x_2)^2 + (y_1 - y_2)^2\| < \varepsilon$, if $f(x_1, y_1) \leq f(x_2, y_2)$,*

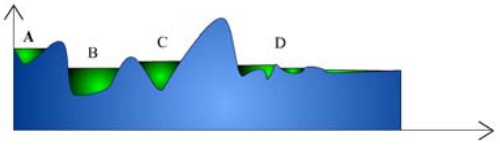
$$f(x_1, y_1) + g(x_1, y_1) \leq f(x_2, y_2) + g(x_2, y_2)$$
$$g(x_1, y_1) \geq g(x_2, y_2)$$
$$g(x_1, y_1) \geq 0, g(x_2, y_2) \geq 0$$

The form of $g(x, y)$ can be trivial for example a zero function. We expect to use $g(x, y)$ to infer the $f(x, y)$. A proper function $g(x, y)$ can have the following three advantages:

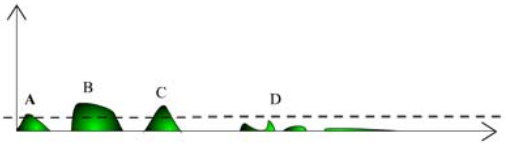
1. **Robustness**. The raw depth data of Kinect sensor contains a lot of noises, and some of the regions may have



(a) $f(x, y)$: original depth image.



(b) $f(x, y) + g(x, y)$: result of water filling.



(c) $g(x, y)$: measure function

Figure 2. Illumination of Water Filling. A,B,C correspond to three people respectively and D is a noise region. Region A has smaller scale compared with B and C, and the absolute hight of A is larger than noise region D. After the water filling process, we get the measure function $g(x, y)$ which reflects the property of $f(x, y)$. Finally the people can be detected by a simple threshold operation on measure function $g(x, y)$.

no depth information, we want to make $g(x, y)$ to be handle this kind of noise.

2. **Locality**. There may be a lot of people in the scene, and different people may have different heights for example adult and kid. To detect people in different heights, the method should have locality property that not influenced by other regions.

3. **Scale-Invariance**. People's head in the view of Kinect sensor may be of different scales, large or small, the function $g(x, y)$ should be scale-invariant so that it can detect the head region at any scale.

We don't need to get a general solution of $g(x, y)$, instead we only need to get a proper non-trivial form of $g(x, y)$. Inspired by the water filling process, we proposed a novel algorithm that can find the proper $g(x, y)$ effectively. The form of function $f(x, y)$ can be seen as a land with humps and hollows. The raindrop in the hump will flow directly to the neighborhood hollow under force of gravity. Little by little, the hollow region will gather a lot of raindrops. The function $g(x, y)$ reflects the quantity of raindrop at (x, y). After the rain stops, the regions with a lot of rain drop can be classified as a hollow.

Our algorithm can be seen as the simulation of the above process. However, there are two problems in simulating the process: one is that, the quantity of raindrop is a continuous value; the other one is that, different raindrops may reach the land simultaneously, their interaction is hard to simulate. In our algorithm we make the following simplifications: 1. every raindrop has the same quantity and the land is discrete; 2. every raindrop reach the land in a order, so that no interaction between two raindrops exists. The detail of the algorithm is shown in Algorithm 1.

Algorithm 1 Inference via Water Filling

Input:
depth image: $f(x, y)$ and measure function $g(x, y) = 0$, with the size of $M \times N$, the threshold T, number of raindrops K.
for k=1:K **do**
  $x = rand(1, M), y = rand(1, N)$
  while True **do**
    $d(x_n, y_n) = f(x_n, y_n) + g(y_n, y_n) - (f(x, y) + g(x, y))$, where (x_n, y_n) is the neighborhood of (x, y).
    $(x', y') = \arg\min d(x_n, y_n)$
    if $d(x', y') < 0$ **then**
      $x = x', y = y'$;
    else
      $g(x, y) = g(x, y) + 1$;
      break;
    end if
  end while
end for
Threshold on $g(x, y)$ with T, and then use contour analysis to find contours and every contour corresponding to a people.
Output: bounding boxes of contours.

The total number of raindrops K is set to be tMN, where t is usually set to be 100. At every loop, (x, y) is randomly generated through a discrete uniform distribution. If there is a point (x', y') in the neighborhood of (x, y) that satisfies $f(x', y') + g(x', y') < f(x, y) + g(x, y)$ then the raindrop in (x, y) flows directly to (x', y') and start another loop until a local minimum is reached. When local minimum is reached, supposing the point is (x_0, y_0), the measure function $g(x_0, y_0) = g(x_0, y_0) + 1$. Since one pixel can only be traveled once for a raindrop, the algorithm can be terminated in finite steps. After all the K raindrops find their stable place, we get the final measure function $g(x, y)$. Due to the locality, robustness, and scale-invariance property, we can use thresholding and contour analysis operation on $g(x, y)$. Besides, the value of $f(x, y)$ in the shoulder is always higher than head, so the drops in the shoulder will find their way to the nearest head region, that make sure there are more drops in the head region than others. The final contours are taken as people (see Figure 2).

Table 1. Experimental Results

Methods	data	Supervised	Accuracy	Recall Rate	F-score
Mean shift	Depth image	No	0.5067	0.8909	0.6460
Random Forest	Depth image	Yes	0.9105	0.8387	0.8731
Water Filling	Depth image	No	0.9916	0.9842	0.9879

Table 2. Generalization Capability on two dataset

	Dataset1			Dataset2		
Method	Recall	Accuracy	F-score	Recall	Accuracy	F-score
Random Forest	0.9105	0.8387	0.8731	0.8759	0.8184	0.8462
Water Filling	0.9882	0.9883	0.9879	0.9947	0.9957	0.9952

3.1. Fast Water Filling

The speed of the proposed water filling algorithm discussed in Algorithm. 1 depends on the number of raindrop which limits the speed in practical applications. In this part, we propose a fast algorithm that can use relatively less raindrops. Our motivation is directly perceived through the sense: raindrops at the neighborhood spot tend to flow to same local minimum regions.

The detailed algorithm is shown in Algorithm 2. Instead of assuming that one raindrop takes one unit water, we make one raindrop take R unit water, where R can usually very big for example 100. In this way, only $K' = K/R$ raindrop is needed to provide the same amount of water thus the number of outer loop is reduced. In the inner loop, the algorithm find regions that lower than its region with a margin r, and then fill the region with r water, where r can be arbitrary value in $[0, R]$. The number of r plays a key role in the algorithm: if r is too big, the algorithm is fast, but the measure function $g(x, y)$ is not fine enough to reflect the property of $f(x, y)$; if r is too small, the algorithm is slow and sensitive to noise. Since Algorithm. 2 is faster than Algorithm. 1 with a margin and achieves similar performance, in the following experiment, we will only use Algorithm 2.

Algorithm 2 Patch based Water Filling

Input:
depth image: $f(x, y)$ and measure function $g(x, y) = 0$, with the size of $M \times N$, the threshold T, number of raindrops K', amount of water in one raindrop R, amount of water dropped one time r.
for $k = 1 : K'$ **do**
  $x = rand(1, M), y = rand(1, N), w = R$
  while $w > 0$ **do**
    $d(x_n, y_n) = f(x_n, y_n) + g(y_n, y_n) - (f(x, y) + g(x, y))$, where (x_n, y_n) is the neighborhood of (x, y).
    $(x', y') = \arg\min d(x_n, y_n)$
    if $d(x', y') + r < 0$ **then**
      $x = x', y = y'$;
    else
      $g(x, y) = g(x, y) + min(r, w), w = w - r$;
    end if
  end while
end for
Threshold on $g(x, y)$ with T, and then use contour analysis to find contours and every contour corresponding to a people.
Output: bounding boxes of contours.

4. Experiment

In this part, we conduct experiments to validate the superiority of proposed vertical Kinect based system and the water filling method in people counting. We compare the detection performance with other state-of-the-art methods, and then we add a tracking module to validate the counting performance.

In order to remove the influence of background, we use GMM(Gaussian Mixture Model) based background modeling [19] as preprocessing and in the following experiment we only operate on foreground regions. In our experiment we collect two datasets in two different scenes. The dataset 1 includes 2834 images with 4541 heads and the dataset 2 includes 1500 images with 1553 heads.

We use three measurements to compare the performance: recall rate, accuracy and F-score, where recall rate is the fraction of people that are detected; accuracy is the fraction of detected result that are people, and F-score is the tradeoff between recall rate and accuracy.

4.1. Experiment on Detection

Since our counting method is based on detection, we firstly compare the performance of our proposed water filling method with the following state-of-the art detection methods:

Mean Shift We use mean shift algorithm [8] to seek the local minimum on depth image. The scale of window is determined according to cross-validation.

Random Forest We use the framework proposed in [18] where depth comparison features and random forest [3]

classifier are used. Similar to [18], we set the number of trees to be 3, and the depth of every tree is set to be 20. For pooling the test result, we use the mean shift clustering.

The experiment is conducted on dataset1. For supervised method, we use the first 1000 image for training and for unsupervised method, we use the first 1000 images for cross validation, the other images are used for test. The experimental results are shown in Table. 1. From Table. 1 we can find that our algorithm outperforms the other two methods. By analyzing the detection results we find the random forest algorithm always fails when multiple people in the scene while mean shift algorithm tends to be sensitive to noise and cannot detect the children. Our proposed water filling algorithm perform very well on both the two situations.

4.2. Experiment on Generalization Capability

To test the generalization capability, we further conduct experiments on dataset2 which is from another scene. We use the model and parameter selected from dataset1. The detection results on dataset2 are shown in Table. 2. Random Forest based method can achieves good performance on the same scenes, but its performance drops on this novel scene. From this experiment, we can find that our water filling method have better generalization capability since that it doesn't rely on the training data.

4.3. Experiment on Parameter Selection

The amount of water dropped one time (r in Algorithm. 2) controls the "smoothness" of measure function, and plays a key role in the algorithm. In this part, we try to examine the impact of this parameter. We conduct experiment on two databases to validate the robustness of the the parameter selection. In all experiment, we set the threshold T as 2 and compare a point with its nearest eight points. The results are shown in Figure 3.

From the experiment we can find that our algorithm is robust since that it achieves relatively good performance when the parameter is selected in the range $[11, 61]$. In the interval the F-score are all above 0.98, which is high enough in practical applications.

4.4. Experiment on People Counting

Since our people detector achieves high accuracy, we can use relatively simple tracking algorithms for people counting. In our system we add a simple nearest neighborhood multiple-target tracking module to the system to get the trajectory of every people and then to count the number. The counting accuracy on dataset2 is shown in Table. 3.

All tested algorithms were implemented in C++, and currently runs at 20 frames per second on a standard PC with Inter Core2 E7500 CPU with 4.0 G memory, and the frame size is 320×240. The speed of the water filling algorithm, is about 30 frames per second. Some examples of the detection and counting results are shown in Figure 4 and Figure 5.

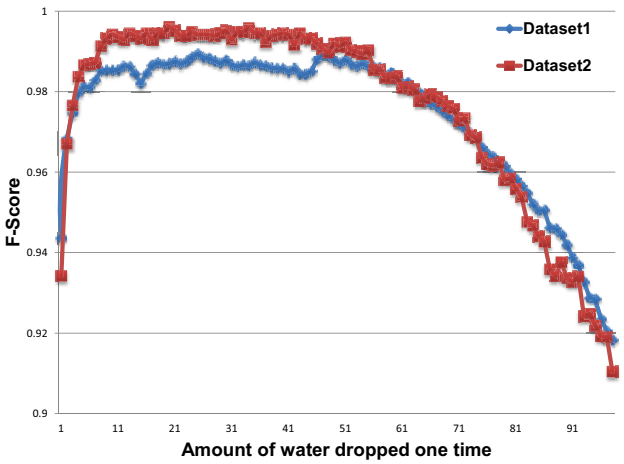


Figure 3. The relationship between the amount of water dropped one time and the F-Score

Table 3. Counting Accuracy on Dataset 2

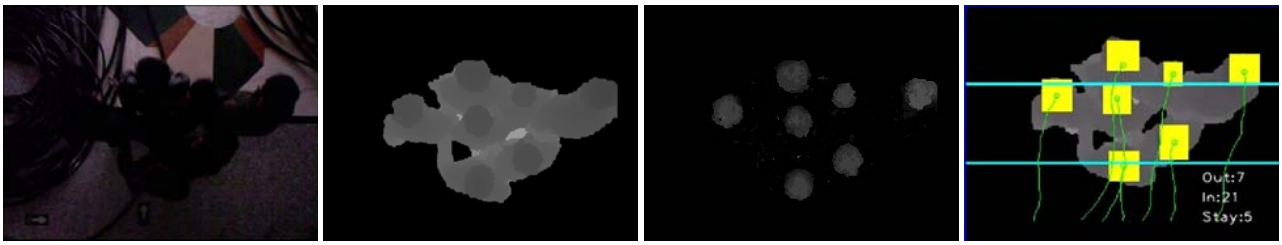
Dataset 1	In	Out	Total
Ground Truth	180	30	210
Water Filling	179	30	209
Accuracy	99.44%	100.00%	99.72%

5. Conclusion

In this paper, we construct a novel people counting system via vertical Kinect sensor which is robust to appearance variations. In the view of vertical Kinect sensor, people counting problem equals to finding the local minimum regions. We propose a heuristic search method that can find these regions and give an patch based version that runs at real time. Experimental comparison with other two algorithms validates the superiority of proposed novel system and water filling inference algorithm, even in the crowed scene and children detection. However, according to the limitation of the algorithm, the water filling cannot handle the situation where some moving object is closer to the sensor than head, such as raising hands over head. In our future work, we will consider to fuse the information of multi-Kinect sensors to cover more large horizon.

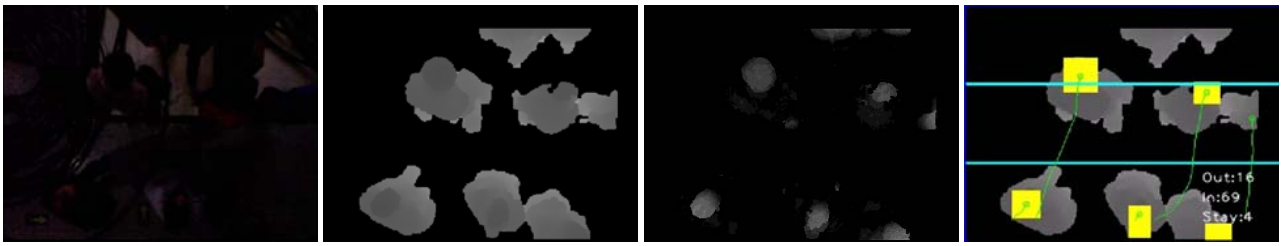
6. Acknowledgement

This work was supported by the Chinese National Natural Science Foundation Project #61070146, #61105023, #61103156, #61105037, National IoT R&D Project #2150510, Chinese Academy of Sciences Project No. KGZD-EW-102-2, European Union FP7 Project #257289



(a) RGB Image (b) Depth Image after background modeling (c) Water Filling Result (d) Detection at Counting Result

Figure 4. Detection and Counting Result. in this frame, all the seven persons are detected.



(a) RGB Image (b) Depth Image after background modeling (c) Water Filling Result (d) Detection at Counting Result

Figure 5. Detection and Counting Result. There are six people in the scene, while only five people detected. One people is missed since that it is partly outside the view.

(TABULA RASA http://www.tabularasa-euproject.org), and AuthenMetric R&D Funds.

References

[1] B. Antic, D. Letic, D. Culibrk, and V. Crnojevic. K-means based segmentation for real-time zenithal people counting. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 2565–2568. IEEE, 2009.

[2] A. Bevilacqua, L. Di Stefano, and P. Azzari. People tracking using a time-of-flight depth sensor. In *Video and Signal Based Surveillance, 2006. AVSS'06. IEEE International Conference on*, pages 89–89. Ieee, 2006.

[3] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[4] G. Brostow and R. Cipolla. Unsupervised bayesian detection of independent motion in crowds. In *CVPR*. IEEE, 2006.

[5] A. Chan, Z. Liang, and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *CVPR*. IEEE, 2008.

[6] A. Chan and N. Vasconcelos. Counting people with low-level features and bayesian regression. *TIP*, 2011.

[7] S. Cho, T. Chow, and C. Leung. A neural-based crowd estimation by hybrid global learning algorithm. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 29(4):535–541, 1999.

[8] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *TPAMI*, 24(5):603–619, 2002.

[9] M. Corp. Kinect for windows. Redmond WA, 2012.

[10] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[11] C. Desai, D. Ramanan, and C. Fowlkes. Discriminative models for multi-class object layout. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 229–236. IEEE, 2009.

[12] L. Dong, V. Parameswaran, V. Ramesh, and I. Zoghlami. Fast crowd segmentation using shape indexing. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[13] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, pages 1627–1645, 2009.

[14] D. Kong, D. Gray, and H. Tao. Counting pedestrians in crowds using viewpoint invariant training. In *BMVC*, 2005.

[15] V. Lempitsky and A. Zisserman. Learning to count objects in images. In *NIPS*, 2010.

[16] A. Marana, L. Costa, R. Lotufo, and S. Velastin. On the efficacy of texture analysis for crowd monitoring. In *Computer Graphics, Image Processing, and Vision, 1998. Proceedings. SIBGRAPI'98. International Symposium on*, pages 354–361. IEEE, 1998.

[17] V. Rabaud and S. Belongie. Counting crowded moving objects. In *CVPR*. IEEE, 2006.

[18] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *CVPR*, volume 2, page 7, 2011.

[19] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, volume 2. IEEE, 1999.

[20] R. Tanner, M. Studer, A. Zanoli, and A. Hartmann. People detection and tracking with tof sensor. In *Advanced Video and Signal Based Surveillance, 2008. AVSS'08. IEEE Fifth International Conference on*, pages 356–361. Ieee, 2008.

[21] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *IJCV*, 63(2):153–161, 2005.