Counting and Tracking People in a Smart Room: an IoT Approach

Dimitris Sgouropoulos, Evaggelos Spyrou, Giorgos Siantikos and Theodoros Giannakopoulos
Computational Intelligence Laboratory
Institute of Informatics and Telecommunications
National Center for Scientific Research – DEMOKRITOS
Athens, Greece
{dsgou, espyrou, dickos, tyianak}@iit.demokritos.gr

***Abstract*—In this paper we present an approach for counting and tracking people participating in a meeting that takes place in a smart room. The sensing and processing modules are incorporated within the context of an IoT framework, that follows a message oriented architecture. The proposed algorithm consists of a motion detection module, a background subtraction module, a people counting module and a tracking module, while its output is primarily used as input to a decision making module that controls the room's environment. We show that we may achieve satisfactory results, using simple low resolution cameras. We evaluate our method using a publicly available, real world data set. Experimental results on real life meetings indicate the effectiveness of our approach.**

I. Introduction

The integration of short range mobile tranceivers into real-life objects, typically not designed to be either connected to the Internet or "computerized", has allowed for new forms of communication among people and objects. Such devices, are typically equipped with unique identifiers and even with processing capabilities and form the concept of the so called "Internet-of-Things" (IoT) [2]. IoT is considered by many as the next industrial revolution. Even if this appears rather optimistic, it is true that a huge number of applications has already been presented in many and heterogeneous areas. Environmental monitoring, infrastructure and energy management, healthcare systems, buildings and home automations are only a few of the research and consumer areas that have benefited from the IoT revolution.

Typical IoT approaches are built around a service-based model. Services are often categorized into three distinctive types: S–type services consist of the raw or slightly processed measurements of sensors, P–type services of the processing of the aforementioned measurements, while the inferred results, decisions and actions are actuated by A–type services. IoT relies on distributed/cloud services which process the measurements gathered by distributed sensor networks. The latter may be restricted over the human body, in small indoor areas (e.g. in a house), or even widespread in large (even world-scale) outdoor areas. These autonomously extracted measurements do not typically require human intervention. Upon the processing, actuation elements may be triggered.

In this work, we present an IoT–based approach for counting and tracking people, in the context of a smart room, at which we have installed several cameras. Our algorithm consists of sensing and processing modules, which exchange messages and does not rely on the selected hardware. For evaluation purposes we collected and annotated a data set of video recordings of real life meetings.

978-1-4673-8395-0/15/$31.00

The rest of this paper is organized as follows: In Section II we attempt to provide an overview of research works that share the same targets, some which are also applied in the context of smart rooms. Then, in Section III we provide the main contribution of this work, i.e. the proposed people counting and tracking scheme. Section IV presents in brief the SYNAISTHISI platform and discusses both hardware and software implementation issues. Experimental results are presented in Section V, while conclusions are drawn in Section VI.

II. Related Work

The problem of counting people has been tackled by many research efforts, using several heterogeneous approaches. However, a study in relevant literature reveals that each approach is in principle motivated by the special conditions of the problem at hand. Many techniques make use of face/person detection algorithms, while others combine blob/object detection and tracking schemes. The problem is typically referred to as "people counting" or "crowd counting" when it is applied to large, open areas. Typical applications focus mainly on surveillance or traffic. The interested reader may find a study focusing on counting in [8].

In order to estimate pedestrian flow counting, Hsieh et al. [6] presented a system that uses Kinect. Their approach was based on morphological processing and extraction of connected components so as to extract regions of interest. They claimed that their system produced real-time results, having perfect accuracy. Ryan et al. [11] proposed an approach that used local features and operated on separate foreground blob segments. This way, they obtained a total crowd estimate as the sum of the group sizes. They claimed that their approach was scalable to unseen crowd volumes, while it required a very small training data set. Zhang et al. [14] presented a system that used a vertically mounted Kinect. They exploited depth information as a means of removing the effect of the appearance variation. They noticed that 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. Thus, they developed an unsupervised water filling method, able to find these regions while being robust and scale-invariant. Zhao et al. [15] used a typical face detection algorithm, enhanced by depth information acquired by a Kinect sensor. Then, they tracked people and counted trajectories. Among their observations we should emphasize the sensitivity of face detection to changing lighting conditions. In the work of Zhao and Nevatia [16], human motion was decomposed into global and limb motion. Global motions were tracked in 3D using ellipsoid human shape models. Zhao et al. [17] defined a joint image likelihood for multiple humans, within the context of a Bayesian framework. This likelihood was formed based on a) the appearance of the humans; b) the body visibility; c) foreground/background separation. In order to find the optimal solution, they used an efficient sampling method, namely the data-driven Markov chain Monte Carlo.

Brostow and Cipolla [3] tracked simple image features and grouped them into clusters which represented independently moving entities with a probabilistic approach. Rabaud and Belongie [10] proposed a highly parallelized version of the well known KLT tracker. A given video was initially processed into a set of feature trajectories. then, a method of spatially and temporally conditioning was applied on the latter. This representation was finally fed to a typical object descriptor. Celik et al. [4] investigated several approaches for perspective distortion correction and proposed a method that relied on foreground object extraction, a perspective correction and a confidence rate that steered a weighted median filter to refine the counts. Ryan et al. [12] proposed a scene invariant crowd counting algorithm, whose goal was to function on multiple calibrated cameras. Features between viewpoints were normalized and regions of overlapping were compensated. They also investigated several features such as object size, shape, edges and keypoints and several regression models such as neural networks, K-nearest neighbors etc. They achieved best results by combining all available features. Chan et al. [5], aiming to protect the privacy of test subjects, proposed a two-step algorithm, where the crowd was segmented into components of homogeneous motion, using the mixture of dynamic textures motion model. Then a set of simple holistic features was extracted from each segmented region. The correspondences between features and the number of people per segment were learned using Gaussian Process regression. This way, they did not use neither object recognition nor tracking. Finally, Yam et al. [13] proposed a real time counting algorithm. They used a surveillance camera with normal mounting. Their technique combined feature matching and point line distance approaches at the object and the detection line. They defined a detection line and counted people that enter or exit.

III. Counting and Tracking People

The aforementioned research works motivated us to design a people counting and tracking scheme robust to lighting conditions. Experiments with face detectors confirmed the observations of [15]. Moreover, since we aimed to use IoT-oriented hardware (i.e. cheap, with limited resources), we ended up with a scheme that combines simplicity with satisfactory real-time results. In this Section we present all parts of the proposed scheme which comprises of a motion detection, a background subtraction, a blob detection and a tracking service.

A. Motion Detection

The motion detection service is the processing element that works as a trigger for the initialization of the proposed system. Its main task is to check the existence of motion within the video stream, at regular time intervals. We define a "motionless state" as a time frame in which practically no movement is detected in our room (to be more precise, we use a relatively small threshold). When the motionless state is detected, the background subtraction service, which shall be described in the next Section, is triggered. During a typical system operation, the reason for which the background subtraction is triggered is that we wish to use each time the best possible estimation of the background. Since certain actions by users (e.g. turning the lights on/off, moving objects/furniture) may significantly alter the background, this choice allows our system to be able to function as intended throughout the day and without the need for human intervention. To conclude, the motion detection service has been designed so as to facilitate two goals: The first is the improvement of the algorithmic aspect of our system, by providing a service initialization trigger and an autonomous reconfiguration scheme. The second is to minimize the use of available resources, i.e. computational power and bandwidth.

B. Background Subtraction

As it has already been discussed, the background subtraction service is triggered from the motion detection service. It captures the current video stream, adjusts the contrast, applies gamma correction and calculates a background model by applying the MOG operator [7]. A set of morphological operations is then applied to further improve the foreground image. First, holes from the estimated foreground model, produced by lighting fluctuations, are erased by a morphological opening. Second, the outline of foreground objects is estimated by performing a morphological gradient. Finally, the remaining black holes existing inside the foreground objects are removed, using a morphological closing operation.

At this step, we expect to have a clear, well–defined foreground image. However, due to the room size and the properties of the camera, foreground objects may lose some information, in terms of their visual characteristics. For example, certain features such as facial features may be discarded due to the aforementioned processing. To overcome this we perform a final step which makes use of a set of region growing algorithms with different expansion schemes. To sum up, the goal of the background subtraction module is to improve the foreground by revealing as much of the foreground objects as possible.

C. Blob (People) Detection

The next and main step of the proposed people counting scheme is a blob detection service. Its input is the foreground image from the aforementioned background subtraction service. It aims to identify image patches (blobs) that in the context of the meeting room, correspond to humans. The algorithm begins by iterating through the pixels from the top-left to bottom-right. This way, the first non–zero pixel that is encountered will always be an area's top left corner. This pixel's position is used to form a rectangle with standard dimensions $N \times N$ and insert it in our list of blobs. The

algorithm then continue by forming rectagles for each new non–zero pixel found and compares each one to the rest of the current list. If the intersection of a newly formed rectangle with one contained in the list, exists and its size is above a predefined threshold, the algorithm proceeds by merging them. Else, the new rectangle is added to the list. When this procedure is over, an additional check is the performed in the list of ROIs for overlaps, in order to merge those that satisfy the aforedescribed conditions.

This process is summarized with the following pseudo-code, where L denotes the list which holds the set of blobs B_i, $i = 1, 2, ..., k$.

Algorithm 1 Blob detection algorithm

```
T ← rectangle union threshold
M ← rectangle merge threshold
for each non-zero pixel p do
    define R ← N × N rectangle
    if L is empty then
        Insert R in L
    else
        if R ∩ B_i > T then
            Insert R ∪ B_i in L
        end if
        for each blob B_i, B_j ∈ L, i ≠ j do
            if B_i ∩ B_j > M then
                B_i ← B_i ∪ B_j
                Delete B_j
            end if
        end for
    end if
end for
```

Finally, a heuristic step is applied to rule out areas that exceed minimum and maximum thresholds. Small areas typically correspond to objects left by the visitors (e.g., cellphones, laptops, keys). On the other hand large areas typically correspond to people standing close to each other so the algorithm proceeds by dividing them accordingly, to reflect their correct number. The standard human size used is an estimation of the dimensions of an average person, given the used camera view, which is set upon an experimental process.

D. Tracking

The last step comprises of a tracking service. Strictly speaking, it stems from the blob detection service. By using the previously mentioned list of blobs we are able to track each individual in our image, i.e. its movement accross it. A seperate list of ids that correspond to the detected blobs is created. Therefore whenever a blob appears or changes (i.e. merged with another blob or divided into two blobs) the list of ids is updated to reflect that change. This enables a very stable tracking scheme with very low computational cost as its main operation is essentially the list update.

IV. Implementation Issues

In this Section we present in brief the SYNAISTHISI platform [1], on which the proposed people counting and tracking approach has been integrated, as a set of sensing and processing services.

A. The SYNAISTHISI platform

The SYNAISTHISI platform has been built in order to a) facilitate the interconnection and the orchestration of a large set of heterogeneous sensing devices; and b) to deliver energy efficient, secure, and effective applications and services to end users. According to their type, services may be distinguished to three categories: a) Sensing (S–) type, which typically consist of physical sensing devices that broadcast their measurements; b) Processing (P–) type, which are in principle software elements that run on a cloud, and their input may consist of the output of S– or other P–type services; and c) Actuating (A–) type, which are triggered by specific P– type services.

The SYNAISTHISI platform is composed of three core systems:

- a *Message Oriented Middleware* (MOM), which is actually a central message broker running on cloud and accessible from all Internet-enabled devices. Its goal is to support inter- and intra-machine communication.
- a *REST web server*, whose role is to provide a control layer over the available resources.
- a *Resource piping mechanism*, which allows for quick development and deployment of custom applications.

All S–, P– and A– services communicate by publishing/subscribing messages within *topics*. The latter are equivalent of addresses. More specifically, measurements, decisions and commands are encapsulated into messages, based on the protocol standard and communicated. Once a message is published on a topic, an additional role of the broker is to inform and deliver the message to all clients that are subscribed to that topic.

MOM is based upon the MQTT protocol [9]. The latter has been designed as an extremely light protocol to allow transport of messages among heterogeneous devices and for applications that require small code footprint and/or network bandwidth. These properties of MQTT along with its availability for many platforms make it very useful for our case, where we have deployed sensing devices that rely on low end hardware and are based on different operating systems and a multitude of programming languages.

Each sensing device incorporates an MQTT client which is connected to the SYNAISTHISI broker that handles the data flow. Since energy efficiency is needed, all sensors are triggered by specific messages, published through MQTT topics. This facilitates scalability, since it allows to add more sensors with almost zero configuration and broadcast commands to all available sensors simultaneously.

The proposed people counting and tracking scheme is part of a smart meeting room. We have transformed a typical meeting room and use it as a testbed for the evaluation of the SYNAISTHISI platform. The corresponding use-case scenario consists of a scheduled meeting within the room, where the ambient state of the room and the number of people present are constantly monitored. Upon a complex event recognition (CER) process, a decision making (DM) module controls the temperature of the room (by powering on/off cooling and/or heating devices) and actuates several devices, such as the lights and the projector. After the end of the meeting, room shuts

down and a speaker diarization process produces transcripted minutes, which are made available to the participants, upon request. The ultimate goals consist of the minimization of the environmental impact, monetary costs, user discomfort, delays and utilization of resources.

To this context, the people counting scheme is a set of S–type services which interoperate, aiming to provide the final measurements to the CER module which shall then decide whether the ambient state of the room is comfort for the specific number of people present and alter it, if needed. Moreover the tracking part aims to assist the speaker diarization service, as it is able to distinguish people present and allow for the extraction and processing of their visual features independently, which may then be fused to the corresponding extracted audio features.

B. Data acquisition and sensor handling

As described in Section IV-A, among the goals of the SYNAISTHISI project is the implementation of low-cost, energy-efficient, easy to use and scalable solutions for various tasks. The proposed setup consists of two Raspberry Pis[1], equipped with Microsoft Kinect cameras[2]. We aimed to build a generic solution, thus we did not use expensive, specialized hardware, nor did we exploit the depth capabilities of the Kinect sensor. Each Pi was programmed using an MQTT client to interconnect with the MOM. The latter was also used to pass the results of any processing service to another and also to fetch the final output, i.e. the number of people counted. Cameras were installed so as to face each other, thus each was capturing one side of the room's table. Thus a complex fusion scheme was unnecessary. Instead, a simple addition was applied. The flow of data as MQTT messages is illustrated in Fig. IV-B.

V. Experiments

A. Dataset

For the sake of the experimental evaluation of the proposed scheme, we have manually annotated videos taken from a real meeting, and captured by two cameras. Figure 2 provides a visual illustration of the output of the proposed algorithm.

The dataset consists of 2 videos (1 from each Kinect), whose resolution is 640×480 and framerate is 7fps. This framerate at first appears low, however it is seriously limited by both network and Raspberry Pi capabilities. The P–type services exchange frames with MQTT messages, via the network infrastructure of our Institution, which is heavily loaded. Moreover, the Raspberry Pis (Model B) used for the sake of our experiments have limited processing capabilities, and do not facilitate a significantly higher frame rate. However, we should remind herein that people counting module is continuously used as input to the DM module, whose ultimate goal is to minimize energy losses, while maintaining certain comfort levels, i.e., improving the overall working environment for room occupants. Thus, the achieved frame rate is more than enough for this application. As for the speaker diarization application, the frame rate appears adequate.

[1]https://www.raspberrypi.org/

[2]http://www.microsoft.com/en-us/education/products/xbox-kinect/default.aspx

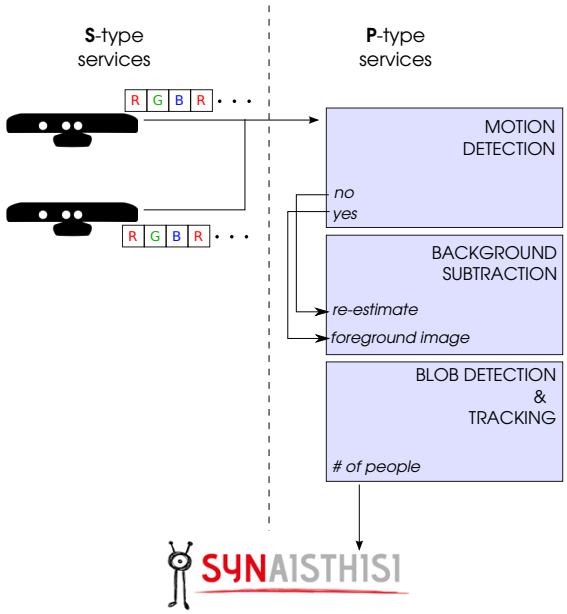

Fig. 1. The flow of data as MQTT messages, of the proposed scheme, within the SYNAISTHISI platform.

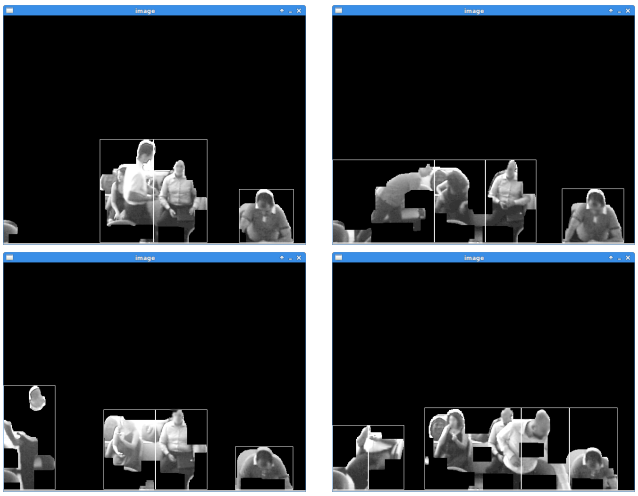

Fig. 2. The visual output of the proposed scheme. Each bounding box corresponds to a counted human.

B. Results

We evaluated the proposed scheme with the following measures: a) the mean absolute error (MAE) as

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N} \left|\#people\_present - \#people\_counted\right| , \quad (1)$$

and b) the mean absolute percentage error (MAP) as

$$\mathrm{MAP} = \frac{100\%}{N}\sum_{i=1}^{N} \frac{\left|\#people\_present - \#people\_counted\right|}{\#people\_present}, \quad (2)$$

where in both cases N denotes the number of observations. A MAE equal to 1.15 and a MAP equal to 23% were achieved. We illustrate the number of people counted by our system vs. the ground truth in Fig. V-B. We should note that the

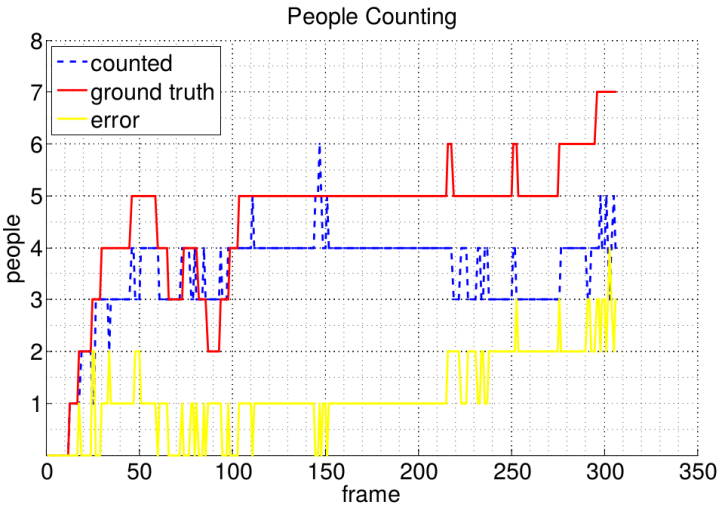

Fig. 3. Counted people vs. ground truth vs. absolute error.

measurements of people counting module are then fuzzified by the CER module as *low*, *normal*, *full*, *overcrowded*, thus the produced errors should be considered relatively small.

VI. Conclusions And Discussion

In this paper we presented a people counting and tracking scheme, tailored for the needs of a smart meeting room. An IoT approach was adopted, thus all sensing and processing modules were built as separate services, exchanging messages. We showed that the proposed scheme achieved good results in real life conditions. Bearing in mind that currently its primary role is to provide an accurate estimate which shall then be fed to a decision making module that aims to control the ambient state of the room, its overall performance is more than satisfactory.

One of the main problems we encountered during the design and the implementation of this approach was to decide which method would better suit our needs. At first we experimented with a system that would rely on face/body detection. However in this case the computational cost was quite high. Moreover, the video resolution needed to provide worthwhile results limited the hardware choices. Our next approach was to perform a preprocessing step to subtract the background and run the face/body detectors on the resulting foreground. While that method worked well in an office environment, it lacked accuracy when used in a meeting room where the cameras were further away and at unfavorable (for this approach) angles. That change resulted to a dramatic drop in the accuracy of the face detector. Our technique was then simplified, under the assumption that within the context of a meeting room moving blobs shall correspond to humans. Upon the application of image correction and optimization techniques, it was further improved, resulting to the proposed one.

Future work shall focus on the inclusion of low-level (e.g. color) features in the tracking scheme. The results of tracking will also be fed to a speaker diarization service. We also plan to add more cameras and apply sophisticated fusion schemes, and also evaluate in larger meeting rooms and/or auditoriums.

Acknowledgment

This research is part of the "SYNAISTHISI" project results. The project is co-financed by the Greek General Secretariat for R&T, Ministry of Education & RA and the European RDF of the EC under the Operational Program "Competitiveness and Entrepreneurship" (OPCE II), in the action of Development Grants For Research Institutions (KRIPIS).

References

[1] C. Akasiadis, E. Spyrou, G. Pierris, D. Sgouropoulos, G. Siantikos, A. Mavrommatis, C. Vrakopoulos and T. Giannakopoulos "Exploiting Future Internet Technologies: The Smart Room Case" In Proc. of International Conference on PErvasive Technologies Related to Assistive Environments (PETRA), 2015.

[2] L. Atzori, A. Iera, and G. Morabito, The internet of things: A survey, Computer networks, vol. 54, no. 15, pp. 2787–2805, 2010.

[3] G.J. Brostow and R. Cipolla. "Unsupervised bayesian detection of independent motion in crowds". In Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, 2006.

[4] H. Celik, A. Hanjalic and E.A. Hendriks. "Towards a robust solution to people counting". In Proc. of Int'l Conf. of Image Processing, IEEE, 2006.

[5] A.B. Chan, Z.-S.J. Liang and N. Vasconcelos. "Privacy preserving crowd monitoring: Counting people without people models or tracking". In Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2008.

[6] C.T. Hsieh,H.C. Wang, Y.K. Wu, L.C. Chang and T.K. Kuo. "A Kinect-based people-flow counting system" In Proc. of Int. Symp. on Intelligent Signal Processing and Communications Systems (ISPACS), IEEE, 2012.

[7] P. KadewTraKuPong and R. Bowden. "An improved adaptive background mixture model for real-time tracking with shadow detection." In Proc. of European Workshop on Advanced Video-Based Surveillance Systems, 2001

[8] C.C. Loy, K. Chen, S. Gong and T. Xiang. "Crowd counting and profiling: Methodology and evaluation". Modeling, Simulation and Visual Analysis of Crowds, pp. 347-382, Springer, 2013.

[9] Mqtt protocol specification. [Online]. Available: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html

[10] V. Rabaud and S. Belongie. "Counting crowded moving objects". In Proc. of Conf. on Computer Vision and Pattern Recognition, IEEE, 2006.

[11] D. Ryan, S. Denman, C. Fookes and S. Sridharan. "Crowd counting using multiple local features". In Proc. of Digital Image Computing: Techniques and Applications (DICTA), IEEE, 2009.

[12] D. Ryan, S. Denman, C. Fookes and S. Sridharan. "Scene invariant multi camera crowd counting". Pattern Recognition Letters vol. 44, pp. 98–112, 2014.

[13] K.Y. Yam, W.C. Siu, N.F. Law and C.K. Chan. "Effective bi-directional people flow counting for real time surveillance system". In ICCE Proceedings, vol. 11, pp. 863-864, 2011.

[14] X. Zhang, J. Yan, S. Feng, Z. Lei, D. Yi and S.Z. Li. "Water filling: Unsupervised people counting via vertical kinect sensor". In Int'l Conf. on Advanced Video and Signal-Based Surveillance (AVSS), IEEE, 2012.

[15] G. Zhao, H. Liu, L. Yu, B. Wang and F. Sun. "Depth-Assisted Face Detection and Association for People Counting". In Pattern Recognition, Springer, 2012.

[16] T. Zhao and R. Nevatia. "Tracking multiple humans in complex situations". IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.26, no.9, pp.1208–1221, 2004.

[17] T. Zhao, R. Nevatia and B. Wu. "Segmentation and tracking of multiple humans in crowded environments". IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.30, no.7, pp.1198–1211, 2008.