IMPROVING THE RELIABILITY OF 3D PEOPLE TRACKING SYSTEM BY MEANS OF DEEP-LEARNING

Matteo Boschini,Matteo Poggi,Stefano Mattoccia

University of Bologna
Department of Computer Science and Engineering (DISI)
Viale del Risorgimento 2, Bologna, Italy
matteo.boschini2@studio.unibo.it,{matteo.poggi8,stefano.mattoccia}@unibo.it

ABSTRACT

People tracking is a crucial task in most computer vision applications aimed at analyzing specific behaviors in the sensed area. Practical applications include vision analytics, people counting, etc. In order to properly follow the actions of a single subject, a people tracking framework needs to robustly recognize it from the rest of the surrounding environment, thus allowing proper management of changing positions, occlusions and so on. The recent widespread diffusion of deep learning techniques on almost any kind of computer vision application provides a powerful methodology to address recognition. On the other hand, a large amount of data is required to train state-of-the-art Convolutional Neural Networks (CNN) and this problem is solved, when possible, by means of transfer learning. In this paper, we propose a novel dataset made of nearly 26 thousand samples acquired with a custom stereo camera providing depth according to a fast and accurate stereo algorithm. The dataset includes sequences acquired in different environments with more than 20 different people moving across the sensed area. Once labeled the 26 K images and depth maps of the dataset, we train a head detection module based on state-of-the-art deep network on a portion of the dataset and validate it a different sequence. Finally, we include the head detection module within an existing 3D tracking framework showing that the proposed approach notably improves people detection and tracking accuracy.

Index Terms— 3D, tracking, stereo vision, deep learning, people detection.

1. INTRODUCTION

Surveillance systems enable video stream analysis for the purpose of gathering meaningful information. Such analysis allows, for example, commercial or marketing studies aimed at enhancing quality of services and other relevant cues. A common scenario in this context is represented by very crowded places, such as shopping malls, where the analysis of the movements, positioning and actions of the customers is crucial in order to improve revenues, retrieve information such as the number of customers, reduce the queues, etc. Other important applications are related to security: being able to visually detect potentially dangerous behaviors allows to prevent potential hazards. People tracking is common to all this tasks and represents a key factor for the success of video analytics and security systems.

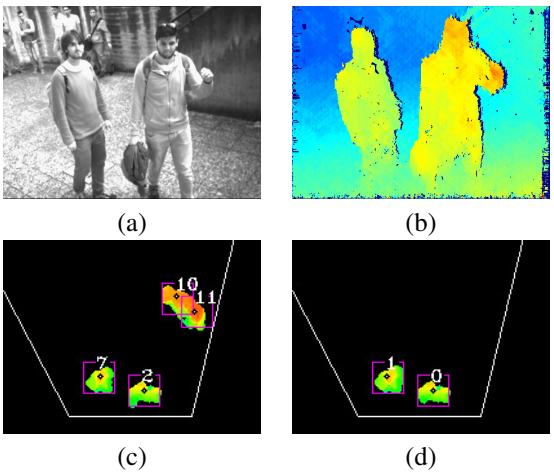


Fig. 1. Overview of the proposed people tracking framework. (a) gray-scale image acquired by a custom RGB-D sensor, (b) disparity map provided by the sensor, (c) *bird view* output obtained by a tracking pipeline from literature, (d) *bird view* output obtained by the same tracking pipeline including the additional deep learning module for head detection described in this paper.

In the past decades a wealth of people tracking techniques have been proposed in the literature. However, most of these approaches are based on 2D information and only a restricted amount exploits depth data, which is actually very easy to obtain by deploying effective and cheap 3D sensors. For several approaches a common setup consists of a down-looking camera (a 2D or a 3D sensor) framing the sensed scene. In order to reduce occlusions and other tracking issues, the camera is often placed at the top of the scene and sometimes with the

image plane almost parallel to the ground plane. However, these setups typically require dedicated infrastructures that might be not always available (for instance in outdoor open areas) or quite expensive to install. Moreover, the sensed area is strictly related to the height at which the sensor is placed. A common alternative setup is represented by a tilted camera with respect to the ground plane, mounted at a lower height; this certainly is more a deployable solution, but is also more prone to occlusions, people hiding and so on. 3D-based systems, can potentially tackle these issues by mapping the problem into a *bird-view* domain 1 according to depth data.

An effective people tracking approach is represented by the *tracking-by-detection* methodology, which models the problem into an object/instance detection problem enhanced with spatial and time constraints. In this paper we aim at improving a 3D tracking system by means of a CNN-based framework in charge of dealing with the people detection step. However, deep learning techniques need a large amount of training data, in particular when deploying very deep architectures. To this purpose, we collected a dataset made of about 26 thousand image + depth frames to train a state-of-the-art object detector. The depth maps (and the reference gray-scale image) are provided by a stereo camera, a sensor suited for indoor and outdoor environments. After labeling the dataset, we trained the deep network for head detection and evaluated the results yielded deploying different cues. Finally, we plugged the trained network within a 3D tracking system working in the *bird-view* domain that typically relies on fragile heuristic parameters for people detection.

2. RELATED WORK

Relevant literature is concerned with visual tracking and object detection. Visual tracking has been extensively analyzed with single and multiple 2D cameras sensing overlapping areas. Li et al. [13] presented a complete survey of the existing visual representations, statistical models, and benchmarks concerning with 2D visual tracking. Despite the large diffusion of such systems, 3D based approaches are potentially more reliable, being able to overcome some of the weaknesses strictly related to purely based 2D methods when dealing with illumination, occlusion issues and so on. Stereo vision is a widely adopted technique to infer depth in this field, in particular when the sensed scene is concerned with an outdoor environment. A first attempt to deal with people tracking using both depth, color and face detection systems has been proposed by Darrell et al. [3]. More effective systems rely on projecting the point-cloud data on the ground plane, mapping the scene into a *bird-view* representation domain, as in [4, 10, 16, 17]. In this domain, these frameworks process common cues such as occupancy and height, in order to detect and track people. Being tracking-by-detection [1] a popular and effective approach, we review relevant literature in the field of object detection, with a particular focus on methods leveraging on deep learning, which proved to be extremely reliable. A very successful machine-learning based method for face detection was proposed by Viola and Jones in [21]. Unfortunately, although very effective, such method would be not suited for head detection purpose, as it is mainly aimed at recognizing faces and not heads, for example, from behind. The first work deploying a CNN in charge of integrated object detection, recognition and localization is Overfeat, by Sermanet et al. [20], deploying a fully convolutional network able to process image of various size and to provide an output of size proportional to the input, according to the pooling operation performed inside the architecture. However, one of the most known method is R-CNN [8], a deep architecture aimed at detection and semantic segmentation by processing a set of proposals, extracted by the input images. Despite the first results achieved by R-CNN were far from real-time processing, follow-up proposal Fast R-CNN [7] and Faster R-CNN [19] moved toward this goal achieving 5 fps. The most recent work concerning object detection by means of deep learning was proposed by Redmon et al. [12], deploying the *You Only Look Once* framework (YOLO). This method proved to be less accurate on standard benchmark such as PASCAL VOC 2012 [6], but extremely faster, achieving a maximum speed rate up to 145 fps. Finally, recent works merged deep learning with visual tracking, in particular Wang et al. [22] analyzed in detail the properties of CNN features in order to exploit them for 2D tracking, Ondruska et al. [18] deployed a Recurrent Neural Network directly mapping raw 2D sensors input to object trajectories in the sensors space without any feature extraction, Zhang et al. [24] showed how a simple two-layer CNN is able to learn a robust representations for visual tracking without off-line training.

3. OVERVIEW OF THE PROPOSED METHODOLOGY

In order to perform 3D people tracking by means of a CNN architecture, we gathered and labeled a large dataset for training and plugged this module within a 3D tracking system based on *bird-view* mapping. Therefore, we propose:

- A dataset for people tracking, and particularly suited for 3D tracking, including about 40K gray-scale and depth images acquired in realistic scenarios by a stereo vision sensor. We labeled a large portion (about 26K) of this dataset in order to allow the training of state-of-the-art deep architectures.

- A 3D people tracking system that leverages on a state-of-the-art object detector based on deep learning and takes as input the 2D image as well as the depth information.

4. DATASET ACQUISITION

In this section, we describe in detail the tools, setup and procedures concerning the acquisition of our dataset. In Section 4.1 we provide details about the RGB-D deployed for the purpose, while Section 4.2 explains the acquisition setup and protocol adopted to create the dataset.

4.1. Custom RGB-D sensor

The dataset was acquired by a custom stereo camera [14] designed for fast and accurate generation of dense depth maps. The camera is slightly tilted with respect to the ground plane as depicted in Figure 2. The sensor processes disparity maps on an on-board FPGA device in real-time at 20+ fps, with a resolution of 640×480. It deploys a full stereo pipeline aimed at *rectification*, *stereo matching*, *outliers detection* and *sub-pixel depth interpolation*. Rectified images are processed through a memory efficient version of the Semi Global Matching algorithm [11] on census transformed images. Sub-pixel interpolation is performed according to a parabola fitting [15], final disparity maps are then filtered according to a speckle filter in order to reduce noise. A further filtering phase is represented by a *low-texture* filter, removing unreliable disparity assignments in low-textured areas and replacing these with undefined values. The stereo deployed for dataset acquisition and for our experiments has relatively a short baseline of about 6 cm. With this configuration, the sensed area is between one and five meters, with a field of view of about $60°$

4.2. Setup and registration

We recorded 9 different sequences, grouped into two dataset.

- **Dataset 1**, acquired in a urban environment, with human-made floor and background. It portrays 16 volunteers crossing the sensed area while performing some common actions, like grabbing a backpack, carrying a helmet, talking to the phone and so. This dataset consists of about 28 thousand frames, grouped into 6 sequences:

  Sequence 1, made of 12765 frames depicting scenes with a single subject walking around and performing common actions or no subjects at all

  Sequence 2, grouping 5931 frames collected when two people walk across the sensed area, performing actions and interacting with each other

  Sequence 3, 3784 frames with three subjects concurrently in the scene

  Sequence 4, 2415 frames including four or more people inside the scene

  Sequence MANY, collecting 1787 samples of very-crowded situations, with about ten people inside the sensed area

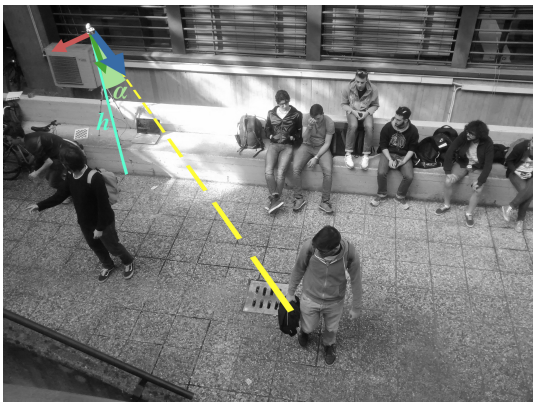


Fig. 2. Setup for dataset acquisition. Custom stereo camera, slightly tilted with respect to the ground plane, located at height h from the ground plane, tilted by an angle α with respect to the mounting support. α is the same for the two datasets and it is equal to $65°$, h changes in the two setups being, respectively, h= 220 cm for dataset 1 and h= 240 cm for dataset 2.

  Sequence OCC, last 1315 frames depicting a single person carrying a large object (a calibration pattern) covering a large portion of his body

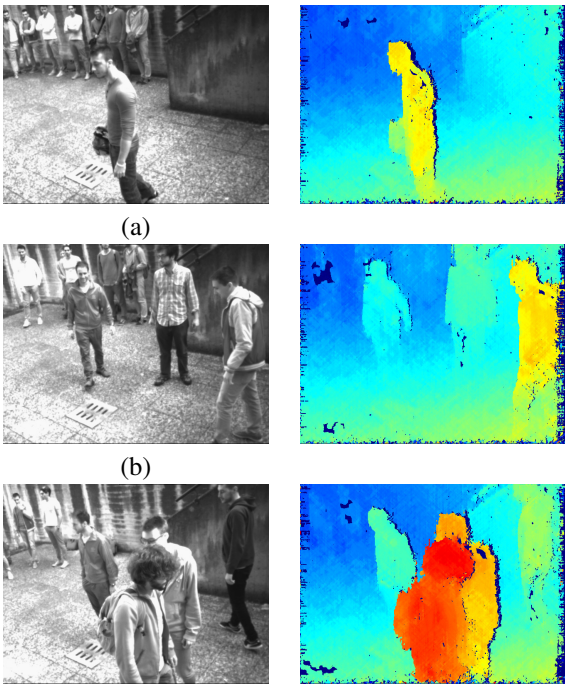
- **Dataset 2**, acquired into a more natural environment, with a mixture of human-made objects (floor, cars, etc) and natural elements (plants) on the background. Seven volunteers move across the sensed area with behaviors similar to those of Dataset 1. It is made of nearly 11 thousand samples, grouped into 3 sequences:

  Sequence 1, made of 9000 frames depicting scenes with a single subject walking around and performing common actions or no subjects at all

  Sequence 2, grouping 1169 images collected when two people walk across the sensed area, performing actions and interacting with each other.

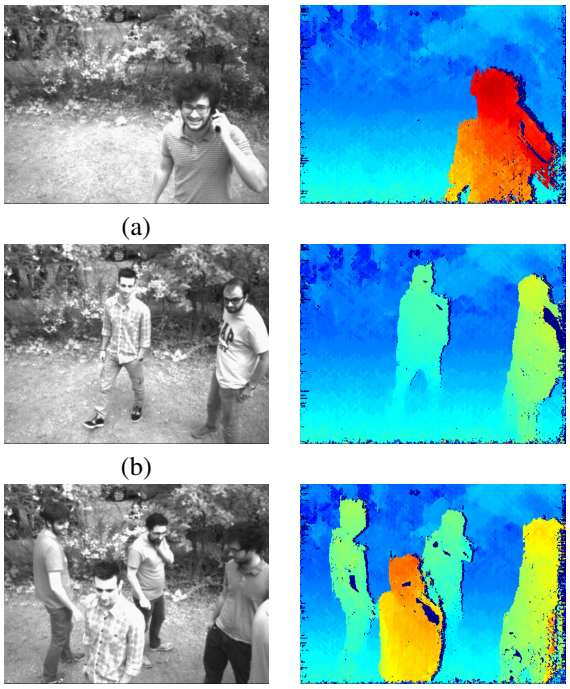
  Sequence MANY, collecting 702 samples depicting situations with 7 people crossing the sensed area

Figure 3 reports a sample for each of the 6 sequences grouped into Dataset 1, showing both gray-scale images and disparity maps encoded in cold colors for farther points and warm colors for closer ones. Similarly, Figure 4 reports samples of the 3 sequences belonging to Dataset 2. A large portion of the acquired datasets has been manually labeled in order to extract ground-truth information concerning the presence of people inside the scene, their position and the associated depth data. The labeling procedure has been carried out in 2D domain by drawing, on each reference image i, a set of bounding boxes B_i centered on the subjects' heads. The depth data is marked according to the same coordinates. Ground-truth data is organized as follow: a single file for each



(a) (b) (c)

Fig. 3. Samples from Dataset 1. On each row, on left we show reference image, on right the disparity map processed by the custom stereo camera (cold colors encode farther points, with lower disparity, while warm colors stand for closer points, with higher disparity). (a) *Sequence 1*, (b) *Sequence 3*, (c) *Sequence 4*.



(a) (b) (c)

Fig. 4. Samples from Dataset 2. On each row, on left we show reference image, on right the disparity map processed by the custom stereo camera (cold colors encode farther points, with lower disparity, while warm colors stand for closer points, with higher disparity). (a) *Sequence 1*, (b) *Sequence 2*, (c) *Sequence MANY*.

sequence s of the 9 collected is provided and formatted as follows:

- *Label file s*:

$$
\begin{aligned}
&i, x_{c_b}, y_{c_b}, w_b, h_b;\\
&i, x_{c_{b+1}}, y_{c_{b+1}}, w_{b+1}, h_{b+1};\\
&...\\
&i+1, x_{c_b}, y_{c_b}, w_b, h_b;\\
&...
\end{aligned}
$$

being i a unique id number for the frame ($i \in [1 : 38868]$), b the index of a bounding box belonging to set B_i, (x_{c_b}, y_{c_b}) the pixel's coordinate of the center of the bounding box i, w_b and h_b the width and height of b.

5. PEOPLE TRACKING

In this section, we provide a description of an algorithm for tracking people moving within the camera's field of view. This algorithm based on the strategy proposed in [4, 10, 16, 17], includes a fast calibration procedure to determine the plane on which the tracked subjects walk and takes advantage of the neural network's predictions in order to reduce the amount of processed data and to make human detection (and, consequently, tracking) more accurate. In particular, our work aims at extending the pipeline outlined in [17].

5.1. Reference system calibration

At start-up, the system initializes a plane detection procedure, in which a single frame coming from the RGB-D camera is analyzed with the purpose of obtaining a reference system that will be used by the tracking algorithm. The disparity map of the scene that is acquired from the sensor is firstly processed to obtain its corresponding 3D point cloud. Using a specific plane detection algorithm, which is based on [23], the largest plane in the scene is found and its equation is obtained as follows:

$$\Pi_g = a_g x + b_g y + c_g z + d_g \tag{1}$$

A reference system with its x and y axes lying on the plane is consequently determined. To infer it, the origin of the camera reference system is projected along the z axis until it meets the detected plane of equation (1) at coordinates

$(0, 0, z_c)$, with z_c obtained by equation (2).

$$c_g z_c + d_g = 0 \tag{2}$$

Then, the first axis is drawn along the intersection of such plane and the yz plane of the camera reference system, along the straight line described by equation (3), the second along the intersection of ground plane with the xz plane of the camera reference system, along the straight line described by equation (4) and the third is obtained by a simple cross product between the two, resulting in a vector $\vec{n} = (a_g, b_g, c_g)$ normal to the ground plane.

$$b_g y + c_g z + d_g = 0 \tag{3}$$

$$a_g x + c_g z + d_g = 0 \tag{4}$$

This calibration method is completely unmanned and extremely fast, thus allowing for an extremely fast configuration and deployment of the system.

5.2. Input Filtering

During tracking, each frame that is captured by the stereo camera is initially processed to obtain a *bird view* map, which is a representation of the scene from a perspective that is orthogonal to the walking plane. In order to do so, the disparity map of the current frame is employed to obtain a point cloud without any form of prior elaboration. The points are then roto-translated according to the coordinate system obtained in Section 5.1. The ground plane is subdivided into square areas/bins and the maps are generated by calculating specific statistics on the points whose projections fall on the same square. Several tracking systems filter the output by means of background subtraction (e.g., [5]), in order to separate foreground from background also greatly reducing the amount input data. Despite that, such technique represents a strong constraint, as a system employing it expects to be given only input where the background does not differ from the one observed as setup-time. For these reasons, we replaced this strategy by means of deep learning based people detection.

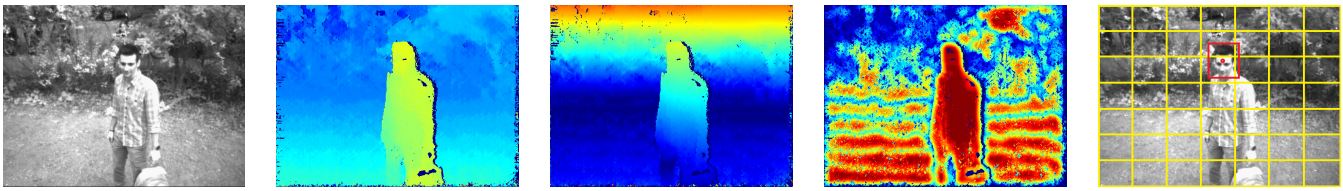
The left two-dimensional image that corresponds to the current frame is contextually submitted to a module in charge of searching for every human head in the picture which yields the positions of their centers. The corresponding points on the disparity map are then projected in the bird view perspective and a circle of radius δ is drawn around each of them thus obtaining a binary mask that will be used to filter out of the bird view maps every point that is too distant from the detected head to assume that it belongs to a person. The people detection approach within a tracking system is primarily aimed at distinguishing humans from objects that have a similar form factor (e.g., coat hangers). The usage of a deep learning-based module for this classification allows our system to be independent of heuristic parameters that describe the appearance of a human being and hence to rely on a highly optimized and effective approach. We deployed a state-of-the-art approach for object detection, the You Only Look Once framework (YOLO) [12], capable of real-time performance on a GPU. It deploys 24 convolutional layers, followed by two final fully connected layers, designed to produce a squared *grid* of size $S \times S$ as output. Each cell of the grid represents a portion of the processed image and contains information concerning the prediction of possible bounding boxes for object detection. A cell can predict up to B boxes, inferring their center, width and height, the category they belong to and confidence (i.e., the reliability of a detection in that box). The output grid is, then, a 3D tensor of dimension $S \times S \times (B \cdot 5 + C)$, with C being the number of categories YOLO has been trained on. In our framework, YOLO performs *head*-detection only, the structure of the network is then adapted to provide the proper outcome, by dividing the image into a 7×7 grid and predicting 2 bounding boxes for each cell. Therefore, YOLO provides a $7 \times 7 \times 10$ tensor, with 10 encoding centers, dimensions and confidence for the two boxes, from which their centers are projected into the bird-view and used to filter the overall map, by removing any point being not inside a radius of δ from any of the centers of the boxes.

Being 3D data available, we encoded it into a meaningful format to be delivered to YOLO. Gupta et al. [9] proposed *HHA* encoding, which consists of building an RGB image, that embeds horizontal disparity, height and angle with gravity direction processed from 3D data into the three image channels. We adopted three encodings, in order to avoid angle computation for each frame, which would require to estimate normals on the entire point-cloud. These proposed encodings are referred to as *LHH*, *LHD* and *HHD*.

- LHH encodes on an RGB image the reference image itself, horizontal disparity and height from the ground.
- LHD replaces horizontal disparity with density, computed by counting, on point-cloud domain, the number of neighboring points inside a radius ρ. This can be achieved efficiently by processing depth maps.
- HHD merges the available 3D features from the previously described formats, excluding the 2D information.

All the features are contained into a range of $[0 : 255]$ values, in particular the maximum considered height encoded is 2.55 m before overflow, enough to distinguish a person from higher/lower elements, while ρ has been set 0.25 m. Figure 5 depicts an overview of the data processed by YOLO and the outcome of its execution.

The statistics that were used to generate separate bird view maps are *occupancy* plus *height* as described in [10, 16, 17] and *color*, which considers the average color for all the points that belong to a patch. The result is finally filtered with a median filter so as to remove isolated points. In our experiments,



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

Fig. 5. Overview of the head detection module. Input data provided to YOLO, respectively (a) 2D image, (b) horizontal disparity, (c) height from the ground, (d) density of the point-cloud. Output from YOLO (e), detecting an head in cell $(4, 3)$. 3D encodings in (b), (c) and (d) are shown in color for clarity, encoding higher values with warmer colors and lower values with colder ones.

Data type	False Pos.	False Neg.	True Pos.	True Neg.	Precision	Recall
2D	0.23	**0.04**	**0.50**	0.23	0.6800	**0.9287**
LHH	0.32	0.12	0.41	0.15	0.5667	0.7769
LHD	0.32	0.10	0.44	0.15	0.5755	0.8147
2D+HHD	**0.08**	0.26	0.27	**0.38**	**0.7628**	0.5117

Table 1. Results concerning the evaluation carried out on Dataset 1, Sequence 2 and Dataset 2, Sequence 1 and 2. The original YOLO approach on 2D data (first row) and the mix of 2D plus HHD encoding (last row) achieve, respectively, superior precision and recall with respect to the other proposals.

we built bird-views encoding bins of size 2×2 cm and radius for filtering of 30 cm.

5.3. Tracking algorithm

The tracking algorithm works on the bird view maps and calculates the position of each person currently in the scene, recognizing the subjects that appear in consecutive frames. For every tracked person, the system stores the following information: status, which may be *tracking*, *candidate* or *lost*, Kalman filter predicting the position and velocity, histograms encoding colors, heights and finally a frame counter to keep track of subject's status. The algorithm is organized in four steps: *prediction*, *measurement*, *localization* and *lost* subjects' matching. During prediction, to each tracked person that is currently recorded in the system is assigned a predicted position obtained with a Kalman Filter by estimating the subject's velocity. Once this phase is complete, the bird view maps for the current frame are examined and blobs of height that are close to a subject's predicted position are recognized, used to update their information within the system and eventually erased from the maps. This recognition relies on a *mean-shift* filtering applied to the height map by iterative repositioning of the kernel window in the position of the center of mass as described in [2]. For this purpose, a simple square kernel window is used whose side's length adapts to the distance between the predicted positions of the subjects. If the recognized subject's status is *candidate*, their frame counter, which represents the amount of consecutive frames in which they were detected, is increased. If the counter exceeds a given threshold, the subject is promoted to the *tracking* status. Conversely, a *candidate* that is not recognized during this phase is removed from the system. Similarly, a *tracking* person who is not recognized is demoted to *lost*.

At the end of measurement, the maps will only contain data that relate to subjects that are not currently tracked. The height map is therefore analyzed in search of values that suggest the presence of a person. Whenever a new subject is found, they are inserted in the system as a *candidate*. In the last step of the algorithm, we try to match the newly found *candidates* with *lost* subjects, comparing their position, their height and their color. If a subject is recognized, it is promoted to the *tracking* status, if it is not, its counter, which represents the number of frames since they disappeared from the scene, is increased. After a reasonable number of frames, a *lost* subject has to be removed from the system, as its position can no longer be accurately estimated.

6. EXPERIMENTAL RESULTS

In this section, we report experimental results concerning the filtering stage addressed by YOLO. In particular, we inquire about the effectiveness of the three different encodings introduced to process 3D data inside the object detector and we compare their results with the plain 2D approach.

6.1. Input filtering

The YOLO framework was fine-tuned for the head detection task by training it on 10094 pictures (i.e., Dataset 1, Sequence 1) using the different image encodings described in Section 5.2. After 19 training epochs, a test was run to de-

(a) (b) (c)

Fig. 6. People Tracking on two frames from Dataset 1, Sequence 1. (a) tracking results without deep-learning based filtering, (b) gray-scale reference image, (c) tracking results with 2D plus HHD YOLO filtering. This latter approach shows to effectively track the subject, while he is hidden by the railing when traditional filtering based on occupancy and height is applied.

termine which training was more effective on a validation set Sequence 2 from Dataset 1, which depicts a the same environment and subjects of the training set, as well as on a testing set made of Sequence 1 and 2 from Dataset 2, acquired in a totally different environment and featuring different subjects. We carry out the evaluation by computing the following statistics:

- **False Positive (FP)**, representing the number of erroneous detections
- **False Negative (FP)**, the number of labeled head YOLO is not able to detect
- **True Positive (TP)**, encoding the number of correctly detected heads
- **True Negative (TN)**, representing the number of frames without heads on which the object detector does not perform erroneous detections
- **Precision** index, defined as the ratio between TP and the sum of TP + FP. It represents the percentage of correct detections among all the detections
- **Recall** index, defined as the ratio between TP and the sum of TP + FN. It represents the percentage of correct detection among the entire set of heads YOLO should be able to label

Table 1 shows the results concerning the validation and testing sets. The table summarizes four configurations: YOLO working on 2D data (i.e., reference image from the stereo camera), on LHH encoded data, LHD encoded data and the joint use of 2D trained YOLO plus a HHD trained one. In this latter configuration, a detection is considered correct if two bounding boxes from the two detectors achieve an Intersection Over Union (IOU) of the area of their bounding boxes higher than 0.1. The table shows how the raw YOLO classifier working on 2D data achieve the best False Negative (the lower the better) and best True Positive (the higher the better) values. The two mixed encodings, LHH and LHD, achieve worse values, suggesting they are not suited for the task. Finally, 2D plus HHD predictions achieve better False Positives (the lower the better) and and True Negatives (the higher the better) with respect to 2D YOLO alone. These numbers highlight how the predictions outcome of both 2D and 3D processing inside YOLO reduces in particular the number of false detections that might occur due to the some ambiguous patterns in the gray-scale domain (e.g., sometimes the bushes from Dataset 2 are recognized as napes) which can be filtered out by analyzing the 3D domain (e.g., density of points in the bushes is different with respect to a human head, etc..). This is reflected into a superior precision index achieved by the 2D plus HHD approach.

6.2. Tracking

Our system proves to be extremely accurate even in situations where a conventional 3D tracking algorithm [4, 10, 16, 17] might fail, thanks to its deep learning-empowered map filtering. In figure 6, we show how it is used to correctly handle an otherwise problematic sequence. In these frames, a railing is indeed present in the background, whose height is compatible with that of a person and therefore it is detected as a row of people by [17]. This misinterpretation of data arises when the tracked subject gets close to the object and hence its identity within the system is confused with the one of the *virtual* subject(s) that form the railing. The introduction of our additional filtering stage completely erases from the maps the railing, which is not detected at all as a human by the CNN, and therefore allows for an accurate and reliable detection of the subjects' position in the scene, even when they get very close to the railing. The proposed people tracking system presents its bottleneck, in terms of execution time, in the head detection module. Nevertheless, the YOLO framework can process a single input in 0.1 seconds on a Tesla C2070 GPU deployed for our tests, enabling a 10 fps output and hence enabling real-time tracking with this GPU and an i7 Intel CPU. On a more recent GPU this time would be significantly lower.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a large dataset made of 40K samples, acquired with a stereo vision sensor, suited for people tracking and deep learning. A large portion, consisting of bout 26K samples, has been manually labeled detecting head locations in 2D and 3D data. The dataset has been deployed for training a state-of-the-art object detector based on CNN

tailored for detecting human heads, whose output is exploited by a people tracking pipeline in order to improve its reliability. Experimental results highlighted how the use of 3D data can improve the precision of the detector. Future work will be concerned with the improvement of the 3D data encodings provided to the CNN, as well as to the deployment of a deep architecture aimed at replacing the tracking module currently based on mean-shift. The full dataset, calibration parameter and label files are available at `http://vision.deis.unibo.it/~mpoggi/datasets.html`.

8. REFERENCES

[1] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.

[2] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603–619, may 2002.

[3] T. Darrell, G. Gordon, M. Harville, and J. Woodfill. Integrated person tracking using stereo, color, and pattern detection. *International Journal of Computer Vision*, 37(2):175–185, 2000.

[4] T.J. Darrell, D. Demirdjian, N. Checka, and P.F. Felzenszwalb. Plan-view trajectory estimation with dense stereo background models. In *ICCV 2001*, pages II: 628–635, 2001.

[5] Luigi Di Stefano, Stefano Mattoccia, and Martino Mola. A change-detection algorithm based on structure and color. In *Advanced Video and Signal Based Surveillance, 2003. Proceedings. IEEE Conference on*, pages 252–259. IEEE, 2003.

[6] Mark Everingham, S. M. Eslami, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vision*, 111(1):98–136, jan 2015.

[7] Ross Girshick. Fast r-cnn. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[8] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 580–587, 2014.

[9] Saurabhand Gupta, Rossand Girshick, Pabloand Arbeláez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. In *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII*, pages 345–360, 2014.

[10] Michael Harville. Stereo person tracking with adaptive plan-view statistical templates. *Image and Vision Computing*, 22:127–142, 2002.

[11] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30(2):328–341, feb 2008.

[12] Redmon Joseph, Kumar Divvala Santosh, Girshick Ross B., and Farhadi Ali. You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Patter Recognition (CVPR)*, 2016.

[13] Xi Li, Weiming Hu, Chunhua Shen, Zhongfei Zhang, Anthony Dick, and Anton Van Den Hengel. A survey of appearance models in visual object tracking. *ACM Trans. Intell. Syst. Technol.*, 4(4):58:1–58:48, oct 2013.

[14] Stefano Mattoccia and Matteo Poggi. A passive rgbd sensor for accurate and real-time depth sensing self-contained into an fpga. In *Proceedings of the 9th International Conference on Distributed Smart Cameras*, ICDSC '15, pages 146–151, New York, NY, USA, 2015. ACM.

[15] V. C. Miclea, C. C. Vancea, and S. Nedevschi. New sub-pixel interpolation functions for accurate real-time stereo-matching algorithms. In *Intelligent Computer Communication and Processing (ICCP), 2015 IEEE International Conference on*, pages 173–178, Sept 2015.

[16] Rafael Muñoz Salinas, Eugenio Aguirre, and Miguel García-Silvente. People detection and tracking using stereo vision and color. *Image Vision Comput.*, 25(6):995–1007, jun 2007.

[17] A. Muscoloni and S. Mattoccia. Real-time tracking with an embedded 3d camera with fpga processing. In *2014 International Conference on 3D Imaging (IC3D)*, pages 1–7, Dec 2014.

[18] Peter Ondruska and Ingmar Posner. Deep tracking: Seeing beyond seeing using recurrent neural networks. In *AAAI*, pages 3361–3368, 2016.

[19] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc., 2015.

[20] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, and Yann Lecun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *ICLR 2014*, 2014.

[21] Michael J. Viola, Pauland Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[22] L. Wang, W. Ouyang, X. Wang, and H. Lu. Visual tracking with fully convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3119–3127, 2015.

[23] J. Xiao, J. Zhang, J. Zhang, H. Zhang, and H. P. Hildre. Fast plane detection for slam from noisy range images in both structured and unstructured environments. In *2011 IEEE International Conference on Mechatronics and Automation*, pages 1768–1773, Aug 2011.

[24] K. Zhang, Q. Liu, Y. Wu, and M. H. Yang. Robust visual tracking via convolutional networks without training. *IEEE Transactions on Image Processing*, 25(4):1779–1792, 2016.