An study on re-identification in RGB-D imagery

J. Lorenzo-Navarro and M. Castrillón-Santana and D. Hernández-Sosa *

Instituto Universitario SIANI
Universidad de Las Palmas de Gran Canaria
Campus de Tafira, 35017 Las Palmas, SPAIN
mcastrillon@iusiani.ulpgc.es

Abstract. Re-identification is commonly accomplished using appearance features based on salient points and color information. In this paper, we make an study on the use of different features exclusively obtained from depth images captured with RGB-D cameras. The results achieved, using simple geometric features extracted in a top-view setup, seem to provide useful descriptors for the re-identification task.

Keywords: re-dentification, surveillance, RGB-D

1 Introduction

There has been an enormous growth in CCTV systems for surveillance in the last fifteen years. The management of the large amount of data acquired justifies the development of automatic surveillance systems that leverage human operator monitoring overload, i.e. the system costs.

Current human monitoring applications focus on non-overlapping camera networks to perform behavior analysis, and automatic event detection. Thus, people detection and tracking approaches are currently being applied in this context aiming at developing automatic visual surveillance systems [1]. A recent application, particularly in automatic monitoring assistance, is the need to re-identify individuals in scenarios with thousands of users (e.g. malls) during a post-analysis of the sequence, after a criminal act, or a dissapearing episode has taken place. Those systems must determine if an individual has been seen in a Time of Interest (TOI) within the camera network.

Facial and clothing information have already been used to re-identify individuals in photo collections and tv video [2]. However, the face pattern presents low resolution in most surveillance systems. Clothing descriptors alone are certainly weak, but can help to locate people with similar appearance, that may be later confirmed by a human. Indeed the human vision system employs external features for person description, body contours, hair, clothes, etc., particularly in low resolution images [3].

Recent literature on the problem of re-identification is mostly focused on appearance based models. Among the appearance cues used for this problem,

* Work partially funded by the Spanish Ministry of Science and Innovation funds (TIN 2008-06068), and the Departamento de Informática y Sistemas at ULPGC.

interest points, structural information and color have deserved researchers attention so far [4–6]. Proving that 2D visual information extracted from RGB images is a valid data source to solve, at least partially, the problem.

However, different authors state the implicit advantage of using depth information to reduce certain ambiguous situations. Thus the design of stereo pair based approaches [7] has been proposed to reduce the inherent illumination problems. But their performance keeps being affected by bad or changing illuminations conditions, as the correspondence map is based on visual information.

Nowadays the Kinect sensor provides affordable rough depth information coupled with visual images. This sensor has already been successfully used to detect individuals, and estimate their body pose. As stated by Harville [8], depth devices: 1) are almost insensitive to shadows and illumination changes, 2) provides additional 3D shape information, 3) include occlusion data, 4) add new types of features to the feature space, and 5) add a disambiguating dimension.

Top view cameras, have already been used in surveillance applications [9], avoiding in many cases an accurate calibration step. This top view configuration has the advantage of being privacy preserving because the face is never grabbed by the camera, being therefore suitable for applications with those restrictions. However, depth information provides new features easy to extract. They lack the distinctiveness to identify uniquely an individual, but provide some evidence that can be used to support or discard a given hypothesis. In our experimental setup, the objective is not to identify precisely any identity, but assist human operators to locate similar individual(s).

In this paper, we study the possibilities to re-identify individuals within the camera network, including depth information in the loop. We claim that current consumer depth cameras can contribute to improve the identity description in the re-identification task.

2 Detection

The aim of this paper is at re-identifying individuals in RGB-D images acquired from a top view setup installed in an entrance door. Individuals are detected based exclusively on the depth cue, using the individual trajectory information to build his/her model.

2.1 Background modeling

Background subtraction is a common technique used to detect objects in surveillance systems. This technique requires a robust background model to be reliable. The solution is particularly simplified if the camera and lighting conditions are fixed, but the model must be robust enough to handle illumination changes. Different approaches to background modeling have been proposed due to its inherent complexity. However, in our scenario, the use of depth information simplifies the segmentation step [8], since considering the top view setup, walking people are clearly salient in the acquired depth images.

(a) (b) (c)

Fig. 1. (a) RGB image, (b) depth image and (c) foreground mask obtained.

We have adopted the background subtraction method proposed by Zivkovic and van der Heijden [10]. According to their approach, a pixel-level background model is built from a Gaussian mixture model (GMM) defined as:

$$p(\boldsymbol{x}|\mathcal{X}_T, bg) \approx \sum_{m=1}^{B} \hat{\pi}_m \mathcal{N}(\boldsymbol{x}; \hat{\boldsymbol{\mu}}_m, \hat{\sigma}_m^2 I) \tag{1}$$

where T is the time window used to estimate the background/foreground model, $\mathcal{X}_T = \{x^{(t)}, \ldots, x^{(t-T)}\}$ is the training set (initial frames), $\hat{\boldsymbol{\mu}}_1, \ldots, \hat{\boldsymbol{\mu}}_B$ are the mean estimations, $\hat{\sigma}_1, \ldots, \hat{\sigma}_B$ are the variance estimations, and I is the identity matrix. For each component in (1), its weight is given by $\hat{\pi}_m$, so if they are sorted in descending order, the number of components B is obtained as:

$$B = \arg\min_b \left(\sum_{m=1} b\hat{\pi}_m > (1 - c_f) \right) \tag{2}$$

where c_f controls the amount of the data that can belong to foreground objects without influencing the background model. Indeed, the number of components in the GMM is not fixed as in other GMM based methods [11].

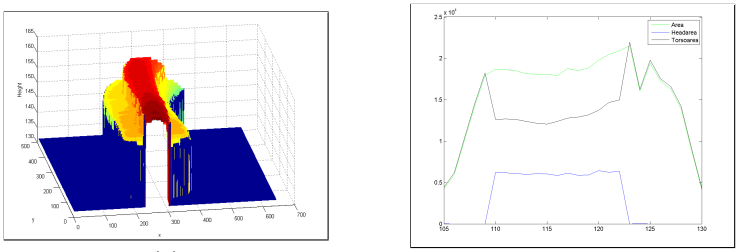
Observing that depth images are less sensitive to shadows and illumination changes, we experimentally determined a value $c_f = 0.2$. The reason for this is that the background model computed for the depth imagery will be much more stable than for RGB images. Given the background model in (1), a pixel belongs to the foreground if the Mahalanobis distance from the pixel value to some component is less than three standard deviations. Otherwise a new component centered in the pixel is generated. Figure 1 shows the background subtraction results for a sample frame along with its corresponding color and depth images.

Thus, according to (1) a pixel $depth(i, j)$ is classified as foreground using the following formula that makes use of a threshold c_{thr} (minimum person height).

$$fg(i, j) = \begin{cases} depth(i, j) & \text{if } p(depth(i, j)|bg) < c_{trh} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

2.2 Tracking

Figure 1c depicts the segmentation results for a sample image based on the depth information. Large connected components in the foreground image are



(a) (b)

Fig. 2. (a) 3D trajectory virtual volume. (b) Area (blob and sub-blobs) related features (in pixels) extracted during a blob tracked trajectory (frames 105-120).

associated to blobs. Given a foreground image fg, the set of m valid blobs is $B = \{b_1, b_2, ..., b_m\}$.

We have adopted a tracking-by-detection approach, based not just on the object bounding box, but its silhouette. Resulting detections are connected in terms of trajectories defined by the tracking.

Tracking is simplified in this top view scenario as occlusions are hardly ever present. As the system is able to acquire and segment on the fly, the blob tracking in frame i can be reduced to test the overlap in consecutive frames of current frame blobs, $B_i = \{b_{i_1}, b_{i_2}, ..., b_{i_m}\}$, and previous frame blobs, $B_{i-1} = \{b_{i-1_1}, b_{i-1_2}, ..., b_{i-1_n}\}$. A new trajectory is considered each time a blob appears in the scene and not suitable matching is found. The trajectory life is then described in terms of its initial and final frames, and the geometric features of its blob and sub-blobs (head and non head, see section 3) components.

3 Modeling

Depth sensors provide a simple mechanism to obtain features that are not trivially computed from the visual cue. As stated before, in this paper we focus on the advantages derived from the use of top view depth images.

Given a foreground image fg, let's define the set of m valid blobs it contains as $B = \{b_1, b_2, ..., b_m\}$. In the case that a blob, b_p, corresponds to a walking human, generally the closest blob point to the camera (lowest gray value), lies on the head. The closest point location and value are useful cues in depth images to split the blob in two parts corresponding to the head and non head areas by a simple in-range operation [9]. Thus, for a given blob, its minimum is defined as:

$$b_{p_{min}} = \min(fg(i,j); \forall fg(i,j) \in b_p) \tag{4}$$

and the head and non head areas as:

$$\begin{aligned} head_p(i,j) &= \begin{cases} fg(i,j) \in b_p \ \wedge \ b_{p_{min}} \leq fg(i,j) \leq b_{p_{min}} \times 1.1 \\ 0 \quad \text{otherwise} \end{cases} \\ nohead_p(i,j) &= \begin{cases} fg(i,j) \in b_p \ \wedge \ fg(i,j) > b_{p_{min}} \times 1.1 \\ 0 \quad \text{otherwise} \end{cases} \end{aligned} \tag{5}$$

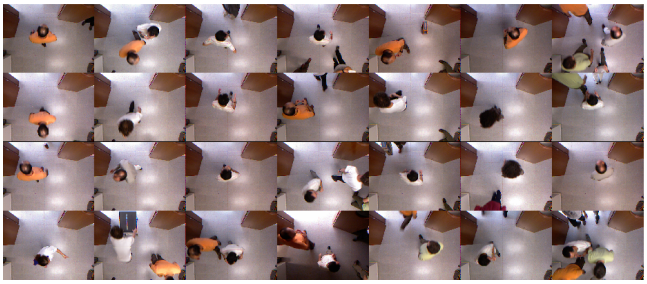


Fig. 3. Middle frame of a subset of the trajectories automatically selected.

The head/no-head split is done whenever the blob container is not too close to the image border. In those situations, the head may be partially or totally out of the field of view and the process leads to erroneous calculations.

Other features may be extracted estimating the individual volume according to the scenario floor. To estimate the floor depth, $depth_{floor}$, it is assumed that that most of the visible area corresponds to the reference floor, i.e. a plane surface. The mean depth image, $\overline{depth}$, is calculated as the average of the k first depth images (assuming that no individual is present) as:

$$\overline{depth}(i,j) = \frac{\sum_{l=1}^{k} depth^{(l}(i,j)}{k} \tag{6}$$

where $depth^{(l}(i,j)$ is the pixel (i,j) of the $l-th$ depth image from the sequence.

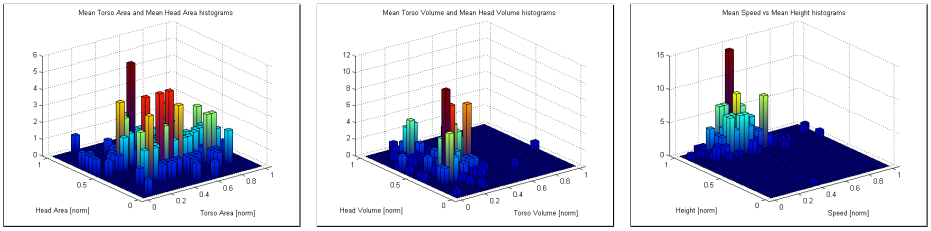
On the resulting $\overline{depth}$, we calculate the mean pixel value to estimate the floor depth, $depth_{floor}$, that is useful to compute the volumetric descriptors:

$$depth_{floor} = \frac{\sum_{i=1}^{height} \sum_{j=1}^{width} \overline{depth}(i,j)}{width \times height} \tag{7}$$

Figure 2a depicts the trajectory of a 3D virtual volume built by means of the successive combination of its tracked blobs. Remember that the depth is given for each pixel, therefore the projected volume can be easily estimated.

After describing the blob subparts and the rough estimation of the scene floor depth, a set of features is defined. For a given set of blobs segmented from the depth image, we have selected the following simple and fast to compute features:

- **Blob height**: Given by the closest to the camera blob point.
- **Blob areas**: The blob and sub-blobs areas (head and non head, if obtained).
- **Blob projected volume**. The blob and sub-blobs (head and non head, if obtained), are projected to the floor. For a blob, b_p, containing $npixels$ pixels, its blob projected volume is computed adding the height value of each blob pixel and subtracting the floor height, $depth_{floor}$, multiplied by the number of blob pixels, i.e. $volume_{b_p} = \left(\sum^{fg(i,j) \in b_p} depth(i,j)\right) - npixels * depth_{floor}$



(a) (b) (c)

Fig. 4. Normalized projection of (a) the area features, (b) the volume features, and (c) the speed and height features for the analyzed trajectories.

The blob tracking defines a trajectory in time. The trajectory features evolve, see for example the area related features shown in Figure 2b. Observe that the head area is not always greater than zero, indeed its value is zero at the beginning and at the end of the trajectory. This effect is due to the fact that when a person enters or leaves the scene, he/she is not completely inside the field of view. Indeed, the head and non head split is only performed when the blob is completely inside the field of view, i.e. its blob container does not "'touch"' the image border. During the trajectory "'middle life"', when the head area is not zero, its features present a fairly constant behavior.

4 Results

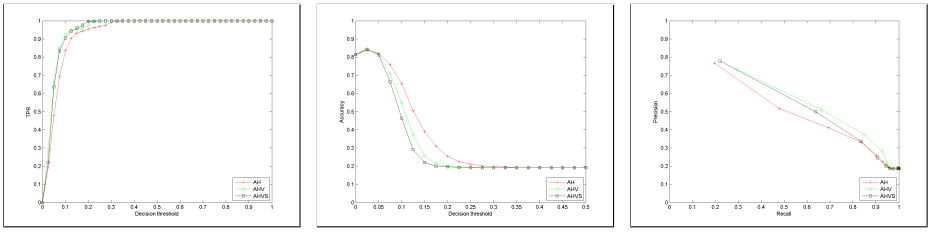
To test the features for re-identification we have collected data using a camera located in the upper frame of a door entrance. The resulting continuous video has been manually annotated, containing around 14200 frames with more than 250 crossing actions with no restrictions imposed to the number of individuals simultaneously present in the field of view, their speed, clothing, etc.

4.1 Trajectory statistics

We have removed those trajectories of individuals not completely visible during the crossing action, so the total number of trajectories analyzed in the experiments is 211. Figure 3 depicts the middle frame of a subset of the total number of trajectories automatically detected by the system. The reader may observe that there are different crossing configurations and illumination conditions.

Histogram based representations of the different features used to describe each trajectory are presented in Figure 4. Each trajectory feature is computed as the mean of the values observed during its "'middle life"', i.e. when the blob could be separated in head and non head sub-blobs.

Figures 4a-b suggest that area and volume information are not coupled. Indeed, two blobs with the same area may project different volumes due to the height difference of the individuals to which the blobs corresponds to. Another consideration should be done related to the speed feature, see Figure 4c. In



(a) (b) (c)

Fig. 5. (a) Recall, (b) accuracy, (c) precision vs recall.

the experiments, some trajectories present particularly high speed. They likely reflect a running crossing action.

4.2 Re-identification

For our re-identification results, each trajectory is compared with the rest in a single-shot approach. This means that we have performed an experiment considering that the training set is composed by a single trajectory features, while the test set contains the rest of trajectories. Thus, the experiment is repeated 211 times for each proposed trajectory representation. The normalized euclidean distance is computed for each test trajectory in relation to the model trajectory. Different feature vectors have been used to describe a trajectory:

- **AH**: Only the area (head and torso) and height features are used.
- **AHV**: The area (head and torso), height and volume (head and torso) features are employed.
- **AHVS**: The area (head and torso), height, volume (head and torso) and speed features are employed.

The lower the distance, the larger the similarity between two trajectories. For each classification problem, the decision threshold defines if a classification is correct or not attending to the distance. The performance evaluation is done using the recall, accuracy and precision. The receiver operating characteristic (ROC) curve is computed for the Nearest Neighbor (NN) classifier considering different decision threshold values. The summarized results are depicted in Figure 5.

As expected, increasing the decision threshold increases the recall or true positive rate, but reduces almost simultaneously the accuracy. The use of more features to describe the trajectory seems to improve the recognition rates. However, the inclusion of the speed feature ($AHVS$ variant) introduce a bias. Certainly, if an individual modifies his speed in different observations, the descriptor is not valid to re-identify him/her.

The results indicate that apparently simple features, provide useful information to re-identify individuals. We can conclude that even using such a simple and weak descriptors the individual re-identification performances are promising. Focusing for instance in Figure 5c, if the decision threshold is set to 0.05

the precision is close to 50% and the recall to 64%. Observe that no appearance based descriptor has been used in the experiments.

5 Conclusions

We have made use only of depth information to detect, track and describe individuals crossing a monitored area. The top view configuration eases the task and makes simple to extract different trajectory features.

No appearance information is used to described the individuals, just geometric descriptors extracted from the blob. Their discriminative power has provided promising results in the experiments.

A set of features has been selected attending to its computational cost. An experimental setup has been carried out in an entrance door scenario, where more than 14000 frames and 250 crossing events have been manually annotated. The selected depth images based features have proven to be useful to detect outliers, and seem to be significant as soft biometrics cues.

References

1. Dollár, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: An evaluation of the state of the art. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(4) (April 2012) 743–761
2. Everingham, M., Sivic, J., Zisserman, A.: Taking the bite out of automated naming of characters in tv video. Image and Vision Computing **27** (2009) 545559
3. Jarudi, I., Sinha, P.: Relative roles of internal and external features in face recognition. Technical Report memo 225, CBCL (2005)
4. D'Angelo, A., Dugelay, J.L.: People re-identification in camera networks based on probabilistic color histograms. In: Proc. SPIE 7882. (2011)
5. Lo Presti, L., Sclaroff, S., La Cascia, M.: Object matching in distributed video surveillance systems by LDA-based appearance descriptors. In: ICIAP. (2009)
6. Muñoz Salinas, R., Aguirre, E., García-Silvente, M.: People detection and tracking using stereo vision and color. Image Vision Computing **25**(6) (2007) 995–1007.
7. Yahiaoui, T., Khoudour, L., Meurie, C.: Real-time passenger counting in buses using dense stereovision. J. Electron. Imaging **20** (July 2010)
8. Harville, M.: Stereo person tracking with adaptive plan-view templates of height and occupancy statistics. Image and Vision Computing **22**(2) (2004) 127–142
9. Englebienne, G., van Oosterhout, T., Krose, B.: Tracking in sparse multi-camera setups using stereo vision. In: Third ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC). (2009)
10. Zivkovic, Z., der Heijden, F.: Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern Recognition Letters **27** (2006) 773–780
11. Stauffer, Grimson: Adaptive background mixture models for real-time tracking. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. (1999) 246–252