Real-time appearance-based person re-identification over multiple Kinect™cameras

Riccardo Satta, Federico Pala, Giorgio Fumera and Fabio Roli
Department of Electrical and Electronic Engineering, University of Cagliari, Italy
{*riccardo.satta, fumera, roli*}*@diee.unica.it, federicopala@email.it*

Keywords: Video surveillance, Person Re-identification, Kinect

Abstract: Person re-identification consists of recognizing a person over different cameras, using appearance cues. We investigate the deployment of real-world re-identification systems, by developing and testing a working prototype. We focus on two practical issues: computational complexity, and reliability of segmentation and tracking. The former is addressed by using a recently proposed fast re-identification method, the latter by using Kinect cameras. To our knowledge, this is the first example of a fully-functional re-identification system based on Kinect in the literature. We finally point out possible improvements and future research directions.

1 INTRODUCTION

Person re-identification is the task of recognizing a person over different cameras, using cues related to clothing appearance (Doretto et al., 2011). In this paper we focus on two issues related to the deployment of re-identification systems in real-world application scenarios: (i) Computational complexity must be low enough to satisfy real-time requirements; (ii) The accuracy of pedestrian detection, tracking and segmentation (first stage of the pipeline) is critical: in particular, accurate segmentation is needed to avoid including background elements. Computational complexity has been overlooked so far in the literature. Among the few exceptions, we mention our previous work (Satta et al., 2012). To improve segmentation accuracy, depth maps provided by range cameras could be exploited (e.g., Time-Of-Flight (Kolb et al., 2009) or code structured light cameras (Salvi et al., 2004)). They could also improve tracking precision, attaining robustness to occlusions and illumination changes, which are difficult to deal with using RGB information only.

We investigated the above issues by developing a fully-functional prototype of a real-time, multiple cameras re-identification system. We exploited the fast re-identification method of (Satta et al., 2012), and the off-the-shelf, low-cost Kinect™range camera, described in Sect. 2. In particular, we used the Kinect free software libraries, that enable enhanced tracking, segmentation and pose estimation, using depth data. We describe the prototype architecture and implementation in Sect. 3, and its evaluation, on a data set we collected in our Lab, in Sect. 4.

2 BACKGROUND

2.1 Fast person re-identification

Most existing re-identification methods subdivide the image $\mathbf{I}$ of an individual into $M \geq 1$ parts $\{I_1, \ldots, I_M\}$ (e.g., torso and legs), and represent each part I_m as a bag (set) of local feature vectors (*components*) $\{\mathbf{i}_m^k\}$ (e.g. patches, SIFT points). The Multiple Component Dissimilarity (MCD) framework (Satta et al., 2012) drastically reduces the matching time of such methods, exploiting the fact that the above representation is likely to exhibit redundancies across individuals that share similar clothing characteristics. MCD turns the above kind of appearance descriptor into a *dissimilarity* one, consisting of a vector of dissimilarities between each body part I_m and a set of representative bag of components for that part, called *prototypes*.

Prototypes are constructed from a given gallery of images of N individuals, $I = \{\mathbf{I}_1, \ldots, \mathbf{I}_N\}$. For each body part $m = 1, \ldots, M$: 1) The feature vectors $\{\mathbf{i}_m^k\}$ of each $\mathbf{I} \in I$ are merged into a set $X_m = \bigcup_{j=1}^{N} I_{j,m}$. 2) X_m is clustered, and the resulting N_m clusters $\mathbf{P}_m = \{P_{m,1}, \ldots, P_{m,N_m}\}$ are defined as the prototypes for the m-th body part. This procedure returns M sets of prototypes $\mathcal{P} = \{\mathbf{P}_1, \ldots, \mathbf{P}_M\}$.

Given the original descriptor of an individual, $\mathbf{I} =$

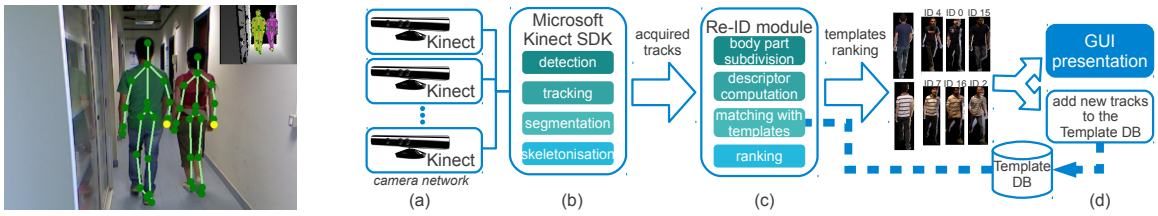

Figure 1: Left: example of the capabilities of the Kinect SDK. Two individuals are tracked, and their skeleton is superimpressed on the image. The upper-right box shows segmentation in depth domain. Right: system architecture (see text).

$\{I_1,\ldots,I_M\}$, the corresponding MCD descriptor is obtained as the concatenation of the M dissimilarity vectors $\mathbf{I}^{\mathrm{D}} = [I_1^{\mathrm{D}},\ldots,I_M^{\mathrm{D}}]$, where:

$$I_m^{\mathrm{D}} = \big(d(I_m,P_{m,1}),\ldots,d(I_m,P_{m,N_m})\big), m = 1,\ldots,M\ ,$$

and $d(\cdot,\cdot)$ is a dissimilarity measure between two sets of components, e.g., the *k-th Hausdorff Distance* (Satta et al., 2012). To match two dissimilarity vectors $\mathbf{I}_1^{\mathrm{D}}$ and $\mathbf{I}_2^{\mathrm{D}}$, a weighted Euclidean distance is used: higher weights are assigned to most significant prototypes (see (Satta et al., 2012) for further details).

A specific implementation of MCD (MCDimpl) was proposed in (Satta et al., 2012). It subdivides body into torso and legs ($M = 2$), and uses as components patches randomly extracted from each body part, described by their HSV colour histogram. Prototypes are constructed using a two-stage clustering algorithm, and are then defined as the patch nearest to the centroid of each cluster. In (Satta et al., 2012) it was shown that MCDimpl can allow several thousands matchings per second, since they reduce to comparing two real vectors. Moreover, although prototype construction can be time-consuming, prototypes can be obtained off-line from *any* gallery of individuals that exhibits a reasonable variety of clothings. In particular, such gallery can be different from the template gallery of the system, and thus does not need to be updated, as new templates are added during operation.

2.2 The Kinect device

The Kinect platform was originally proposed for the home entertainment market. Due to its low cost, it is currently gaining much interest over the computer vision community. The device provides: (i) an RGB camera (1280×960 pixels at 30fps); (ii) an IR depth sensor based on code structured light, which constructs a 640×480 pixels depth map at 30fps, with an effective range of 0.7 to 6 meters. The Kinect SDK also provides reliable tracking, segmentation and skeletonisation (Shotton et al., 2011), based on depth and RGB data (see Fig. 1-left).

The technology adopted by the Kinect device suffers from two limitations. First, the maximum distance at which a person can be detected (around 5–6 mt) is relatively low. However, ad hoc sensors (probably more costly) can be developed to deal with higher distances, based on the same technology. Second, the use of IR projectors and sensors to build the depth map prevents outdoor usage, because of the interference in the IR band caused by the sun light. Indoor environments include nevertheless typical video-surveillance scenarios (e.g., offices, airports).

3 SYSTEM IMPLEMENTATION

Our prototype tracks all the individuals seen by a network of Kinect cameras, adds to a template data base an appearance descriptor (*template*) of each acquired track, and re-identifies online each new individual, by matching its descriptor with all current templates. After a track is acquired, the operator is shown the list of templates, ranked according to the matching score to that track.

Our prototype architecture is shown in Fig. 1 (right). It consists of a network of Kinect cameras, connected to a PC (Fig. 1, right (a)). First, detection, tracking, segmentation (i.e., silhouette extraction) and skeletonisation of each individual seen by the network are carried out (Fig. 1, right (b)), exploiting the Kinect SDK (other detection techniques based on RGB and range data can also be used, e.g., (Salas and Tomasi, 2011; Spinello and Arras, 2011)). Each individual is associated to a *track*, i.e., the sequence of regions of the RGB frames containing him/her, extracted by the detector, and the corresponding skeletons. After a track T is acquired, a template is created by the re-identification module (Fig. 1-right(c)), and is added to the Template DB. A template is made up of the acquisition date and time, and of the 5 frames $\{T_1,\ldots,T_5\}$ of the track exhibiting the largest silhouette area, with the corresponding MCDimpl descriptors.

Online re-identification is performed for each new track T, with respect to all current templates. The first frame of T is initially matched to the templates, and then subsequent frames sampled every $t_{\mathrm{acquire}} = 1$ sec., but only if the corresponding silhouette area

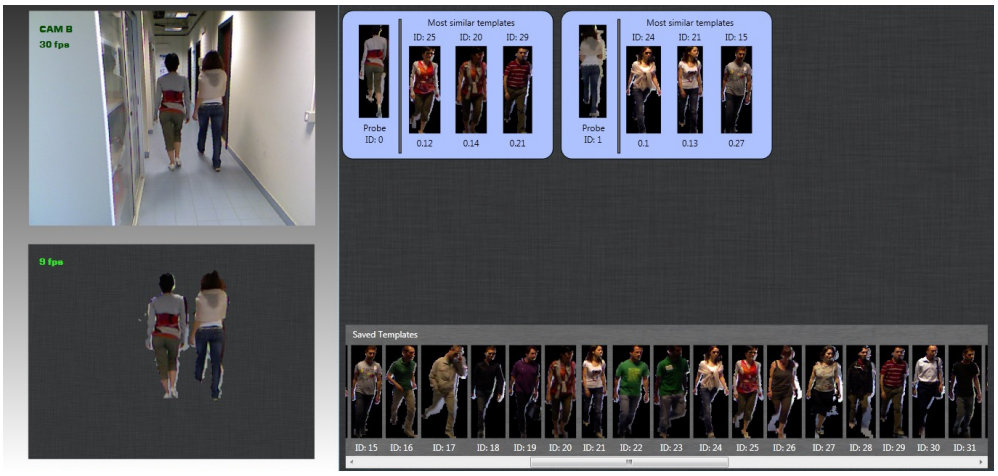

Figure 2: Screenshot of the main view of the prototype GUI, showing images coming from one camera. The user can run many of these views to check simultaneously different cameras. Left: RGB output (top) and segmentation (bottom). Right: the best frame (i.e., the one with the largest silhouette area), the matching score and the template ID of the three template tracks most similar to the one on the left (top); the current template tracks (bottom).

is greater than the one of the previously processed frame. For a given frame f to be matched: (1) the silhouette is subdivided into torso and legs; (2) an appearance descriptor q is created; (3) q is matched to all current templates. If a frame contains up to two individuals, the Kinect SDK provides their skeleton, made up of 20 different joint points. In this case, in step (1) the torso is obtained by taking all pixels between the *hip center*, and of the middle point between the *head* point and the *shoulder center*. Similarly, the legs body part is obtained from the pixels between the *hip center*, and of the lowest foot. If the skeleton is not available, the body is subdivided using the method of (Farenzena et al., 2010).

Steps (2) and (3) rely on MCDimpl (see Sect. 2.1). In our system the prototype gallery was built off-line, using the 1.264 pedestrian images of the VIPeR data set (Gray et al., 2007), which exhibit a wide variety of clothing appearances. The matching score between q and a template track T was computed as the median score between q and the MCDimpl descriptors of the best frames $\{T_1, \ldots, T_5\}$.

For each new track, the sorted list of templates (Fig. 1(d)) is shown to the operator through a Graphical User Interface (Fig. 2), where one representative frame for each template is displayed. By clicking on it, the other frames are also displayed.

4 EVALUATION

We evaluated our system in a real scenario: a small network of two Kinect cameras, A and B, deployed in an office building (see Fig. 4, left). The software was installed on a PC with a dual-core i5 2.3GHz CPU. We first collected a data set of videos of 54 people wearing different clothing. Two video sequences were acquired for each individual: frontal (camera A) and back pose (camera B). This is a challenging data set, due to strongly different poses and non-uniform illumination (see Fig. 4, middle). It is available upon request. The experiments were carried out by injecting the video sequences into the system, simulating their online acquisition. The video sequences of camera A were used to populate the Template DB: in this step, re-identification was disabled. Then, re-identification was performed for each video-sequence of camera B, without subsequently adding them to the Template DB. The above experiment was repeated by switching the roles of cameras A and B.

Re-identification accuracy was assessed via the Cumulative Matching Characteristic (CMC) curve, i.e., the probability of finding the true identity among the first n templates, as a function of n, averaged over the two runs (Fig. 4, right). Note that the reported performance takes into account all processing steps, from tracking to matching, while most previous works focus only on descriptor creation from silhouette, and matching. Computing one descriptor from a frame (including all preprocessing steps) took about $t_{\text{descr}} = 50$ ms. Matching one probe with one template track took about $t_{\text{match}} = 0.03$ ms.

The maximum number I of individuals that can be tracked and re-identified simultaneously is approximately $I = \frac{t_{\text{acq}}}{t_{\text{proc}}}$, where t_{proc} is the processing time required to re-identify one frame. It is given by $t_{\text{proc}} = t_{\text{descr}} + M \cdot t_{\text{match}}$, where M is the number of tracks in the Template DB. In Fig. 3 we plotted I vs

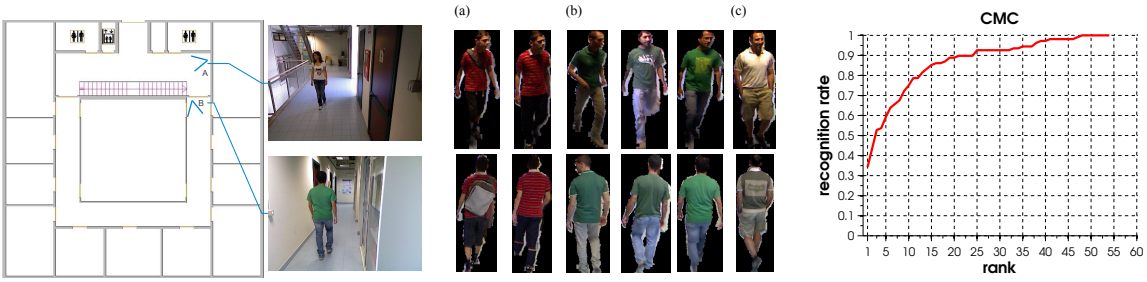

Figure 4: Left: position of cameras A and B. Middle: examples of front and back views from our data set: (a) the same individual wearing different clothing and cumbersome accessories; (b) different people wearing clothing of the same colour, and (c) clothing with strong differences between front and back view. Right: average CMC curve attained by our prototype.

M using $t_{\text{acq}} = 1$ sec. (see Sect. 3), it can be noted that M influences I only slightly, due to the very low t_{match} of MCDimpl. In real scenarios, M depends on the number of individuals seen by the camera network in the period of time of interest (e.g., one day). In small indoor scenarios (e.g., office buildings), a reasonable value of M is 400-500. The actual number of distinct individuals may be much smaller, but it is likely that each one would be seen many times by the camera network. In this case, around 15 different individuals can be re-identified simultaneously by our system. Even in larger environments ($M \geq 1000$) the value of I decreases only slightly. It is worth noting that a security operator is usually interested in re-identifying only one or a few individuals at a time. Thus, the performance of the proposed system should already satisfy such needs, in small/medium indoor scenarios.

5 CONCLUSIONS

We described the implementation of an online re-identification system working with multiple Kinect cameras, which addresses the challenging issues of computational complexity, and of tracking and segmentation. To this aim, we used the fast re-identification method of (Satta et al., 2012), which enables on-line re-identification over a camera network, and carried out tracking and segmentation exploiting the depth map provided by the Kinect device, although its technology limits its application to indoor environments. Two main improvements can be foreseen. First, temporal reasoning on the spatial layout of the camera network can be exploited, to avoid matching the current track with templates acquired distantly in space or time. Second, a quality measure should be defined to select the most representative frame of a track, for template construction. We used to this aim the silhouette area, but lighting conditions and body pose could be considered as well.

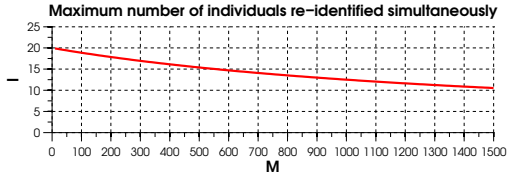

Figure 3: Maximum number I of individuals seen by the camera network that can be re-identified simultaneously, versus the number M of template tracks.

ACKNOWLEDGEMENTS. This work has been partly supported by the project CRP-18293 funded by Regione Autonoma della Sardegna, L.R. 7/2007, Bando 2009.

REFERENCES

Doretto, G., Sebastian, T., Tu, P., and Rittscher, J. (2011). Appearance-based person reidentification in camera networks: problem overview and current approaches. *J. Amb. Intell. and Human. Comp.* 2:127–151.

Farenzena, M., et al. (2010). Person re-identification by symmetry-driven accumulation of local features. In *CVPR, 2010*.

Gray, D., Brennan, S., and Tao, H. (2007). Evaluating appearance models for recognition, reacquisition, and tracking. In *PETS, 2007*.

Kolb, A., Barth, E., Koch, R., and Larsen, R. (2009). Time-of-Flight Sensors in Computer Graphics. In *Eurographics 2009 - State of the Art Reports*.

Salas, J. and Tomasi, C. (2011). People detection using color and depth images. In *MCPR*, 2011.

Salvi, J., et al. (2004). Pattern codification strategies in structured light systems. *Patt. Rec.* 37(4):827–849.

Satta, R., Fumera, G., and Roli, F. (2012). Fast person re-identification based on dissimilarity representations. *Patt. Rec. Lett.* 33(14):1838–1848.

Shotton, J., et al. (2011). Real-time human pose recognition in parts from single depth images. In *CVPR*, 2011.

Spinello, L. and Arras, K. (2011). People detection in rgb-d data. In *IROS*, 2011.