Appeared in International Workshop on Information Quality
and Quality of Service for Pervasive Computing, 2011

Video Entity Resolution: Applying ER Techniques for Smart Video Surveillance

Liyan Zhang Ronen Vaisenberg Sharad Mehrotra Dmitri V. Kalashnikov

Department of Computer Science
University of California, Irvine

***Abstract*—Smart Video Surveillance (SVS) applications enhance situational awareness by allowing domain analysts to focus on the events of higher priority. This in turn leads to improved decision making, allows for better resource management, and helps to reduce information overload. SVS approaches operate by trying to extract and interpret higher "semantic" level events that occur in video. On of the key challenges of Smart Video Surveillance is that of *person identification* where the task is for each subject that occur in a video shot to identify the person it corresponds to. The problem of *person identification* is very complex in the resource constrained environments where transmission delay, bandwidth restriction, and packet loss may prevent the capture of high quality data. In this paper we connect the problem of person identification in video data with the problem of *entity resolution* that is common in textual data. Specifically, we show how the PI problem can be successfully resolved using a graph-based entity resolution framework called RelDC that leverages relationships among various entities for disambiguation. We apply the proposed solution to a dataset consisting of several weeks of surveillance videos. The results demonstrate the effectiveness and efficiency of our approach even with low quality video data.**

***Keywords*-Smart Video Surveillance, Video Entity Resolution, Person Identification, Video Data Cleaning**

I. Introduction

Advances in sensing, networking, and computational technologies has created the possibility of creating sentient pervasive spaces wherein sensors embedded in physical environments are used to monitor its evolving state. There are numerous physical world domains in which sensors are used to enable new functionalities and/or bring new efficiencies including intelligent transportation systems, reconnaissance, surveillance systems, smart buildings, smart grid, and so on.

In this paper, we focus on smart surveillance systems wherein video cameras are installed within buildings to monitor human activities. Such a surveillance system could support variety of tasks such as building security to new applications such as locating/tracking people, inventory, etc., or tasks such as analysis of human activity in shared spaces (such as offices) to bring improvements on how the building is used. One of the key challenges in building smart surveillance systems is that of automatically extracting semantic information from the video streams. Such semantic information may correspond to human activities, events of interest, etc. that can then be used to create a representation of the state of the physical world (viz. building). Such a representation, when stored inside a sufficiently powerful spatio-temporal database can be used to build variety of monitoring and/or analysis applications. Most of the current work in this direction focuses on computer vision techniques. Automatic detection of events from surveillance videos is a difficult challenge and the performance of current techniques, often leaves a room for improvement. While event detection consists of multiple challenges, (e.g., activity detection, location determination, etc.), we focus on a particularly challenging task of *person identification*.

The challenge of *person identification* (PI) consists of associating each subject that occurs in the video with a real-world person it corresponds to. In the domain of computer vision, the most direct way to identify a person is to perform face detection followed by face recognition, the accuracy of which is limited even when video data is of high quality. Thus, in the resource constrained environments, where transmission delay, bandwidth restriction, and packet loss may prevent the capture of high quality data, face detection and recognition becomes more complex. We have experimented with Picasa's face detector on our video dataset (704×480 resolution/frame), and found that it can detect faces in only 7% of the cases.

In this paper, we explore a new approach leveraging contextual data, such as time, space, and activities to improve the performance of person identification. Our approach is proved to be efficient even with low quality data, because contextual features are more robust in presence of video data with poor quality. To exploit contextual information for PI, we connect PI problem with a well-studied entity resolution problem [2], [5], [9], which is typically considered in the context of textual data. Entity resolution is a very active research area where many generic approaches have been proposed, many of which could potentially be applied to the PI problem. In this paper, we apply a relationship based approach for entity resolution (which we refer to as RelDC) developed in [7], to the PI problem. RelDC is an algorithm and framework for analyzing object features as well as inter-object relationships, to improve the quality of data cleaning. In this paper we will demonstrate how our RelDC framework for entity resolution could be leveraged to solve a person identification problem that arises when analyzing video streams produced by cameras installed in the CS Department at UC Irvine.

The rest of this paper is organized as follows. Section 2 presents RelDC framework for entity resolution. Section 3 describes how to map person identification problem into RelDC's input. Section 4 demonstrates experiments and results. Section 5 concludes this paper.

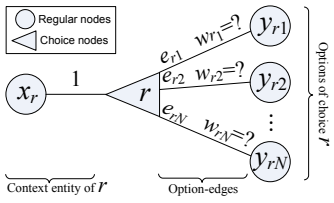

Fig. 1. Choice Node.

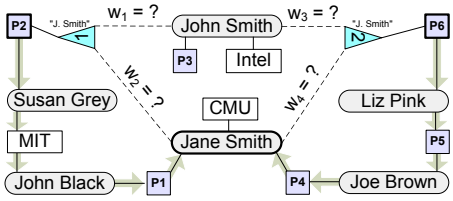

Fig. 2. Examples of Choice Nodes.

II. RelDC Framework for Entity Resolution

A. ER Problem Definition

In the general setting of the Entity Resolution problem, the dataset contains information about the set of objects $O = \{o_1, o_2, \ldots, o_n\}$. The objects in the dataset are represented by the set of their descriptions $R = \{r_1, r_2, \ldots, r_m\}$, such that each object is represented by one or more descriptions. Entity Resolution has two main instances: *Lookup* [2], [5] and Grouping [2], [9].

We primarily will be interested in an instance of the lookup problem which is defined as follows. Let $X = \{x\}$ be the set of all entities in the dataset. Each entity x consists of, among other things, a set of references $\{r_1, r_2, \ldots, r_n\}$. Each reference r is essentially a description of some entity. For instance, in a publication scenario, publications contain references to authors. The entity, in the context of which the reference r is made, is denoted x_r. The set of all references in the database is denoted as R. Each reference $r \in R$ semantically refers to a single specific entity in X called the *ground truth* for reference r and denoted g_r. The description provided by r may, however, match a set of one or more entities (*options* for r) in X. We refer to this set as the *option set* of reference r and denote it by S_r. The option set consists of all the entities that r could potentially refer to: $S_r = \{y_{r1}, y_{r2}, \ldots, y_{r|S_r|}\}$. To simplify notation, we will use N to mean $|S_r|$, that is $N = |S_r|$. In general, for r its g_r is unknown to the algorithm, and the goal is to find it.

B. Relationship-based Data Cleaning

Over the past few years we have developed a powerful disambiguation engine that we refer to as the *Relationship-based Data Cleaning* (RelDC) [3]–[6], [8], [10]. In this section we outline the principal methodology employed by the RelDC framework. RelDC works by representing and analyzing datasets in the form of Entity-Relationship (ER) graphs. In such graphs, entities are represented as nodes and edges correspond to relationships among entities. In RelDC, the ER graphs are augmented further to represent ambiguity in data. Such an augmented graph is then analyzed to discover interconnections, including indirect and long connections, between entities which are then used to make disambiguation decisions to distinguish between same/similar representations of different entities as well as to learn different representations of the same entity. RelDC is based on a simple principle that entities tend to cluster and form multiple relationships among themselves. Before we summarize how the framework can be applied to the problem of person identification, we first in the subsequent sections illustrate RelDC methodology as applied to the Lookup problem. The discussion will ignore many details and intricacies for the sake of brevity. Further details can be found in [5], [8]. Application of RelDC to the Grouping problem is described in [3], [4].

Entity-Relationship Graph. RelDC views the dataset being analyzed as an undirected entity-relationship graph $G = (V, E)$, where V is the set of nodes and E is the set of edges. The set of nodes V is comprised of the set of regular nodes V_{reg} and the set of *choice* nodes V_{cho}. Each regular node corresponds to some entity $x \in X$. If for r its S_r has only one element y_{r1}, then r is resolved to y_{r1}, and graph G contains an edge between x_r and y_{r1}. If S_r has more than 1 element, then a choice node is created for reference r, as illustrated in Figure 1. This reflects the fact r is ambiguous and that g_r can be one of $y_{r1}, y_{r2}, \ldots, y_{rN}$. Given the direct correspondence between a reference r and its choice node, we will use the same notation r for both of them.

Node r is linked with node x_r via edge (x_r, r). Node r is also linked with N nodes $y_{r1}, y_{r2}, \ldots, y_{rN}$, for each y_{rj} in S_r, via edges $e_{rj} = (r, y_{rj})$ for $j = 1, 2, \ldots, N$. Edges $e_{r1}, e_{r2}, \ldots, e_{rN}$ are called the option-edges of choice r. The weights of option-edges are called option weights. Each weight w_{rj} of edge e_{rj} for $j = 1, 2, \ldots, N$ is undefined initially: they are variables, the values of which are to be determined by the disambiguation algorithm. Since option-edges $e_{r1}, e_{r2}, \ldots, e_{rN}$ represent mutually exclusive alternatives, the constraint is added that their weights must add up to 1: $\sum_{j=1}^{N} w_{rj} = 1$.

The goal of Lookup is to resolve all references as correctly as possible, that is, for each reference $r \in R$ to correctly identify g_r. RelDC achieves that by first computing the weights w_{rj} for each $y_{rj} \in S_r$, which reflect the degree of its confidence that y_{rj} is g_r. It then *interprets* those weights to disambiguate all references, by resolving each $r \in R$ to y_{rj} where $j = \arg\max_\ell w_{r\ell}$. Figure 2 illustrates the concepts discussed in this section. It shows a publication scenario, where two of the publications P_2 and P_6 have uncertain references to authors. In both cases the references specify the author as only 'J. Smith', which is ambiguous and can correspond to either 'John Smith' or 'Jane Smith'. The uncertainty is captured by the two choice nodes, marked as '1' and '2'.

Connection Strength. The concept of Connection Strength is at the core of the proposed RelDC approach. The connection strength $c(u, v)$ between the two nodes u and v reflects how strongly these nodes are related to each other via relationships in the graph G. The value of $c(u, v)$ is computed according to

some connection strength *model*. Logically, the computation of $c(u, v)$ can be divided into two parts: first finding the connections, and then measuring the strength in the discovered connections.

In general there can be many connections between nodes u and v in G. Intuitively, many of them (e.g., very long ones) are not very important. To capture most important connections while still being efficient, instead of discovering all paths, the algorithm discovers only the set of all L-short simple paths $\mathcal{P}_L(u, v)$ between nodes u and v in graph G. A path is L-short, if its length is no greater than parameter L. A path is simple, if it does not contain duplicate nodes. Because of certain semantics of the approach, some of the discovered paths are illegal, and they are ignored by the algorithm. Finding connections is the bottleneck of the overall approach.

To measure the strength of the discovered connections RelDC uses a connection strength model, see [5] for an overview. For instance, some of such models compute the connection strength of path p as the probability of following path p in graph G via random walks. Many of the existing models compute $c(u, v)$ as the sum of the connection strengths of paths in $\mathcal{P}_L(u, v)$:

$$c(u, v) = \sum_{p \in \mathcal{P}_L(u,v)} c(p). \tag{1}$$

These models differ in the way they compute $c(p)$.

Optimization Problem. Given the connection strength measures $c(x_r, y_{rj})$ for each unresolved reference r and its options $y_{r1}, y_{r2}, \ldots, y_{rN}$, we can use the CAP to determine the desired weights w_{rj}. Note that CAP does not contain any specific strategy on how to relate weights to connection strengths. Any strategy that assigns weight such that if $c_{r\ell} \geq c_{rj}$ then $w_{r\ell} \geq w_{rj}$ is appropriate, where $c_{r\ell} = c(x_r, y_{r\ell})$ and $c_{rj} = c(x_r, y_{rj})$. In particular, we use the strategy where weights $w_{r1}, w_{r2}, \ldots, w_{rN}$ are proportional to the corresponding connection strengths: $w_{rj} c_{r\ell} = w_{r\ell} c_{rj}$. Using this strategy and given that $\sum_{j=1}^{N} w_{rj} = 1$, the weight w_{rj}, for $j = 1, 2, \ldots, N$, is computed as:

$$w_{rj} = \begin{cases} \frac{c_{rj}}{\sum_{j=1}^{N} c_{rj}} & \text{if } \sum_{j=1}^{N} c_{rj} > 0; \\ \frac{1}{N} & \text{if } \sum_{j=1}^{N} c_{rj} = 0. \end{cases} \tag{2}$$

Thus, the model for computing $c(p)$, Eqs. (1) and (2), and the paths that exist between x_r and y_{rj} in G, define each option weight w_{rj} as a function of other option weights $\mathbf{w}$: $w_{rj} = f_{rj}(\mathbf{w})$:

$$\begin{cases} w_{rj} = f_{rj}(\mathbf{w}) & (\text{for all } r, j) \\ 0 \leq w_{rj} \leq 1 & (\text{for all } r, j) \end{cases} \tag{3}$$

The goal is to solve System (3). System (3) might be over-constrained and thus might not have a solution. Thus, similar to SVM, a slack is added to it by transforming each equation $w_{rj} = f_{rj}(\mathbf{w})$ into $f_{rj}(\mathbf{w}) - \xi_{rj} \leq w_{rj} \leq f_{rj}(\mathbf{w}) + \xi_{rj}$. Here, ξ_{rj} is a slack variable that can take on any real nonnegative value. The problem transforms into solving the optimization problem, where the objective is to minimize the sum of all ξ_{rj}:

$$\begin{cases} \text{Constraints:} & \\ f_{rj}(\mathbf{w}) - \xi_{rj} \leq w_{rj} \leq f_{rj}(\mathbf{w}) + \xi_{rj} & (\text{for all } r, j) \\ 0 \leq w_{rj} \leq 1 & (\text{for all } r, j) \\ 0 \leq \xi_{rj} & (\text{for all } r, j) \\ \\ \text{Objective: Minimize } \sum_{r,j} \xi_{rj} & \end{cases} \tag{4}$$

System (4) always has a solution and it can be solved by a solver or iteratively. The result of solving it are the values for all w_{rj} weights.

Interpretation Procedure. As the final step, when resolving reference r and deciding which entity among $y_{r1}, y_{r2}, \ldots, y_{rN}$ from S_r is g_r, RelDC chooses such y_{rj} that w_{rj} is the largest among $w_{r1}, w_{r2}, \ldots, w_{rN}$. Once this is done, the outcome of the disambiguation can be used to create a regular database. An interesting property of the proposed solution is that it is *global*, as it finds the overall combination of all weights that is consistent with the overall connection strength and fits the system best, rather then making small local decisions at a time.

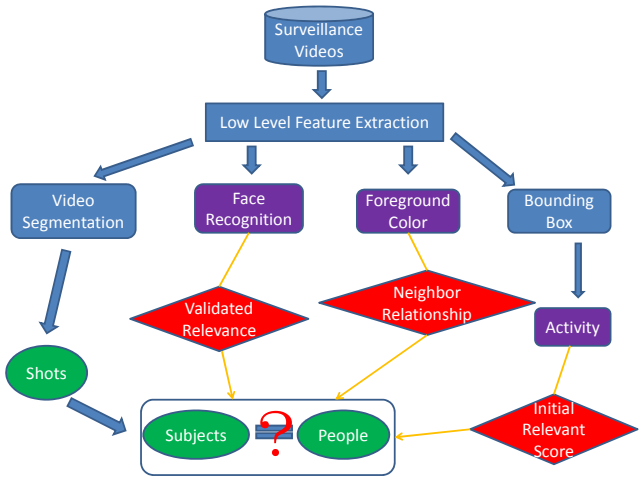

Fig. 3. Workflow.

III. Mapping PI Problem into RelDC's Input

In this section we discuss how person identification problem can be mapped into the graphical representation which RelDC framework can take as input and perform disambiguation. The overall process is illustrated in Figure 3.

A. Low Level Feature Extraction

1) Temporal Segmentation: Temporal segmentation is an essential part in video processing. We segment videos into *shots*. Intuitively, subjects appearing in consecutive frames are likely to be the same person. Hence, we initially group frames into shots just based on the time continuity. But time continuity alone can not guarantee person continuity. If the subjects' color histograms of two consecutive frames are significantly different, they will not be placed into the same shot. After video segmentation, we obtain a series of shots. For each shot, we select one frame to represent it. Usually, we choose

a middle frame that contains whole and large image of the subject.

2) Foreground Color Extraction: Foreground color composition, determined primarily by person clothes, is one of the essential features to identify a person. Although people may change clothes across different days, having the same clothes during the same day is a strong evidence that two images contain the same person. To accurately capture the color information of an individual in the image, we separate the person from the background by applying a background subtraction algorithm [1]. After color extraction processing, the foreground area is represented by a 64-dimensional vector, which consists of a 32-bin hue histogram, a 16-bin saturation histogram, and a 16-bin brightness histogram.

3) Face Detection and Recognition: Face detection and recognition is a direct way to identify a person. However, it does not perform well in our dataset due to several reasons. First, the surveillance cameras used are of low quality. The resolution of each frame is 704×480. Second, people may actually walk away from cameras, in which case the cameras only capture their backs and not faces. Because of that, the best face detection algorithms we have tried could only detect faces in about 7% of frames, and recognize 1 or 2 faces for a frequently appearing person. Although the result is not ideal, we could still leverage it for further processing. We define a function $FR(x_i, p_j)$ which reflects the result obtained by the face recognition. If x_i and p_j are the same according to face recognition, we set $FR(x_i, p_j) = 1$, and otherwise $FR(x_i, p_j) = 0$.

4) Bounding Box and Centroid Extraction: To track the trajectory of an object and obtain activity information, we need to extract bounding box and centroid of objects. To do that we apply a simple computer vision algorithm. Given three consecutive frames with the same object, we first obtain the differences of the first two frames by subtraction, and then acquire the differences of the last two frames. By combining the two different parts, we get the location of objects. After obtaining the bounding box, we determine the centroid of subjects by averaging the points of x-axes and y-axes.

B. Event Detection

Higher-level events can be detected based on the extracted low-level features. We extract events such as walking direction and entering a room. Such events prove very relevant to the problem of person identification. For example, entity entering an office is a very strong signal about its identity: it is likely to be either (one of the) person(s) who works in this office or their collaborators and friends. Similarly, the trajectory and walking direction can serve as a weak, but useful, signal indicating the identity of the individual.

1) Walking Direction: The most common activity in dataset is walking. The walking direction (towards or away from the camera) is an important factor to predict the subsequent behavior of a person. Walking direction can be obtained automatically by analyzing the changes of the centroid between two consecutive frames in a shot. For example, if noticing that the centroid of subject is moving from the bottom to the top in the image, we could determine that this person is walking away from the camera.

2) Activity Detection: We focus on detecting simple regular type of behavior of people, including entering and exiting a room, walking through the corridor, standing still, and so on. These types of behavior can be determined by analyzing the bounding box of the first and last frame in a shot, which we will refer to as *entrance* and *exit* frames. By analyzing the bounding box (BB) of a subject in the entrance frame, we could predict where the subject has come from. Similarly, the exit frame could tell us where this person is headed to. If we consider all the BBs in entrance and exit frames, we could easily find several locations in the image, where people are most likely to appear or disappear. These locations, denoted as $L = \{l_1, l_2, \ldots, l_{|L|}\}$, can be automatically computed in an unsupervised way by clustering the centroid of entrance/exit BBs. Based on this analysis, we could automatically obtain the entrance and exit point in an image.

After computing the set of entrance and exit locations $L = \{l_1, l_2, \ldots, l_{|L|}\}$, we can compute the distance between them and determine the entrance and exit points in each shot. Suppose that in a shot the subject walks from location l_p to l_q, then we can denote the activity as $act_i : \{l_p \rightarrow l_q\}$.

C. PI Problem Formalization

In the previous sections, we have segmented videos into shots, extracted low level features (color, faces, bounding boxes), and detected events (e.g., which locations the subject moves from/to in the shot). We now will show how to represent the problem as an entity resolution problem for RelDC.

1) Notation: Let *D* be the surveillance video dataset which stores a large amount of videos with motion. Let $S = \{s_1, s_2, \ldots, s_{|S|}\}$ be the set of all shots in D. Each shot s_i consists of a set of properties/features that we extracted on the previous steps. Each shot consists of several consecutive frames, from which we can select one (middle) frame to represent this shot. The time of this frame t_i is used to represent the time of shot s_i, and the subjects $\{x_{i1}, x_{i2}, \ldots\}$ appearing in this frame are considered to represent all subjects of the whole shot. For each subject x_{ij} its color histogram H_{ij} is computed. In addition, the activity information $act_i : \{l_p \rightarrow l_q\}$ for shot s_i is extracted, which means people in shot s_i walk from location l_p to location l_q.

Let $P = \{p_1, p_2, \ldots, p_{|P|}\}$ be the set of (known) people of interest that appear in our dataset. By applying face detection and recognition we can match some of the subject x_i to the corresponding person p_j, in which case we set $FR(x_i, p_j) = 1$. Although face recognition succeeds in only 7% of the cases, we can still employ $FR(x_i, p_j)$ as a hint to match the remaining subjects with people via transitive closure. The goal is for each subject x_i to identify the person p_j it corresponds to or output `other`.

2) Entity-Relationship Graph: We construct an entity-relationship graph $G = (V, E)$, where V is the set of nodes and E is the set of edges. Each node corresponds to an entity and each edge to a relationship. There are several types of nodes that correspond to *shot*, *subject*, *person*, *color*

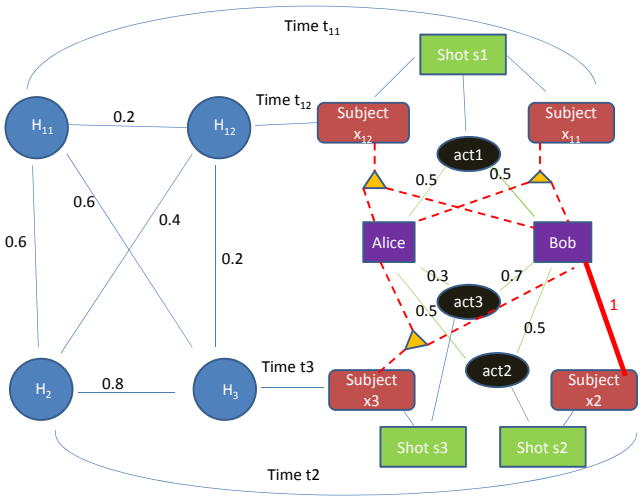

Fig. 4. Example of Entity-Relationship Graph

histogram and *activity* information. The edges linking these nodes corresponds to the relationships. For instance, the edge between a shot node and a subject node correspond to the "appears in" relationship.

In graph G, edges have weights where a weight is a real number in [0,1] that reflects the degree of confidence in the relationship. For example, if there is an edge with weight 0.8 between a subject node and a person node, this implies the algorithm has 80% confidence that this subject and person are the same. The edge weight between two color histogram nodes denotes their similarity.

Figure 4 illustrates an example of an entity-relationship graph. It shows a case where the set of people of interest consists of just two persons: Alice and Bob. It considers three shots s_1, s_2, s_3, where s_1 captures two subjects x_{11} and x_{12}, shot s_2 captures x_2 and s_3 has x_3. The goal is to match people with shots. Subject x_{11}, x_{12}, x_2, x_3 in the graph are connected with their corresponding color histograms H_{11}, H_{12}, H_2, H_3. An edge between two color histogram nodes represent the similarity between them. For instance, the similarity of H_2 and H_3 is 0.8. In addition, subjects are connected to the corresponding activities, which could be indicative of who these subjects are. For example, if past labeled data is available, from the fact that subject s_3 is connected to activity act_3, we can get the prior probability of 0.7 that s_3 is Bob. The graph also shows that according to face recognition subject x_2 in shot s_2 is Bob.

3) Color histogram similarity: A color histogram is a 64-dimensional vector. For two histograms H_i and H_j taken on the same day we compute their similarity based on their Euclidean distance. However, if two histograms are take on two different days, we set the similarity to zero. The similarity is used as the edge weight between the nodes of two histograms.

4) Determining Person Based on Activity: By using event detection, we have extracted activity information for a subject in a shot, such as entering a room, exit a room and etc. Intuitively, we might be able to predict a person by the activity information. For example, if a person enters or exits Bob's office frequently, there is high probability this person is Bob. In general, given label past data we can compute priors

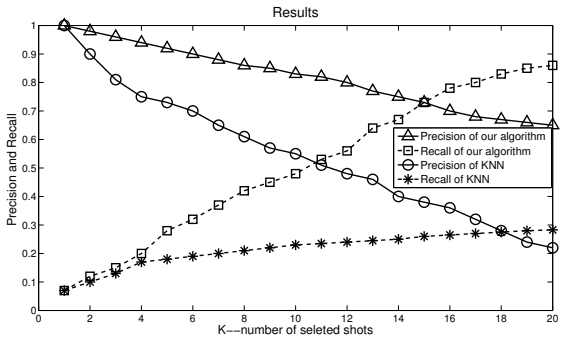

Fig. 5. Experimental Result

such as $\mathbb{P}(p_j|a_i)$, which corresponds to the probability that the observed subject is p_i, given that the subject participates in activity a_i, such as entering/exiting a certain location l_p. Similarly, we can compute $\mathbb{P}(p_j|a_i, t_k)$ which also considers time.

D. Applying RelDC

After applying RelDC, we get a relevance score $R(x_i, p_j)$ for each pair of subject x_i and person p_j. The higher $R(x_i, p_j)$ is, the more likelihood the subject x_i is person p_j. Using $R(x_i, p_j)$ we can rank all the scores related to person p_j, and set top-ranked subjects to be person p_j.

IV. Experiment

We have collected 2 weeks' surveillance videos from 2 adjacent cameras in the CS Building of UC Irvine, captured at 1 frame/second when motion is detected. The resulting video shots are relatively simple, with one (or, rarely, a few) person(s) performing simple activities. The task is to map the unknown subjects into known people.

To test the performance of the proposed algorithm, we manually labeled 4 people from the video dataset to assign the ground truth labels. The video collected over 2 weeks contains several (over 50) individuals of which we manually labeled 4. We then have divided the dataset into 2 parts. The first week has been used as training data and the second week as test data. From the training data, we get the faces of the chosen 4 people, and train a face recognizer. We also extract activities of people, and compute priors based on activities.

We have applied RelDC (in a limited form with a simplified connection strength model) to identify the four people from the testing dataset. After obtaining the relevance scores $R(x_i, p_j)$, for each p_j we get the set of top-K subjects according to the score. For each subject x_i in this set we use its $R(x_i, p_j)$ to decide which p_j to assign it to: $j = \text{argmax}_j\, R(x_i, p_j)$. From the training data, we know that there are about 10-20 shots for each person every day. Thus we set $1 \leq K \leq 20$, and compute precision and recall for the first K shots. Figure 5 illustrates the average precision and recall achieved by the proposed approach for the four people, with K ranging from 1 to 20. We observe that with the increase of K, precision decreases and recall increases. The average F-measure reached by the proposed algorithm at $K = 20$ is 0.76. We also run a KNN

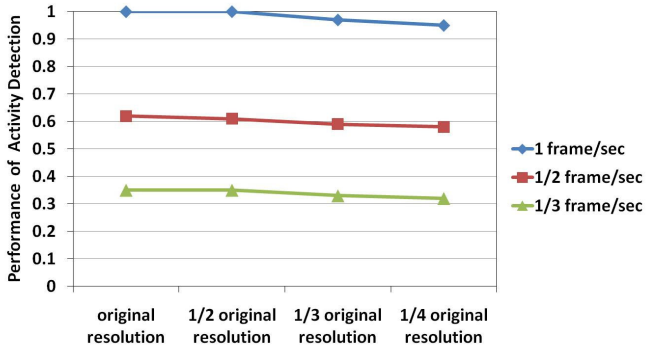

Fig. 6. Activity Detection with Decreasing of Resolution and Sampling Rate

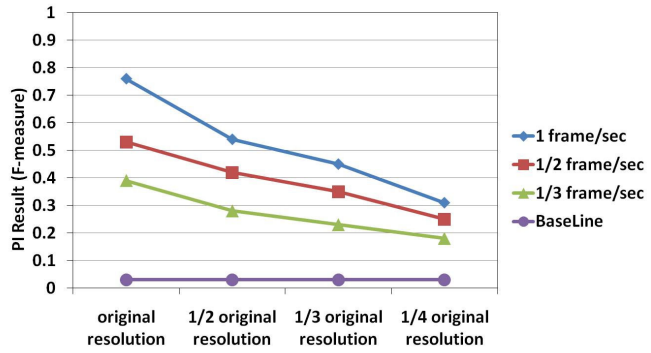

Fig. 7. PI Result with Decreasing of Resolution and Sampling Rate

clustering which employs only color information as a baseline and that method gets 0.24 F-measure. The result demonstrates that the proposed algorithm is significantly better than this simple baseline. This is since it considers activities and their relationships instead of just relying on color histogram.

To test the robustness of our approach, we degrade the resolution and sampling rate of frames in our dataset respectively, and run a series of experiments on such dataset. From Figure 6, we can see that the performance of activity detection (suppose the performance with the original resolution and sampling rate is 100%) drops when sampling rate reduces from 1 frame/sec to 1/2 and 1/3 frame/sec, because many important frames are lost with the decrease of sampling rate, however, the decrease of resolution does not affect the performance of activity detection since the contextual information (such as time and location) does not change. Figure 7 illustrates that person identification result (F-measure when $k = 20$) drops with the reduction of resolution and sampling rate, due to the lost of activity and color information. However, PI result of our algorithm even with the lowest resolution and sampling rate is much better than the baseline results of Naive Approach (which predicts results just based on the occurrence probability in the training dataset). Consequently, Figure 7 demonstrates the robustness of our approach with low quality video data, because our approach leverages contextual data rather than merely relying on the quality of video data.

V. Conclusion

In this paper we considered the task of person identification in the context of Smart Video Surveillance. We have demonstrated how an instance of indoor PI problem (for video data) can be converted into the problem of entity resolution (which typically deals with textual data). The area of entity resolution has become very active as of recently, with many research groups proposing powerful generic algorithms and frameworks. Thus, establishing a connection between the two problems has the potential to benefit the PI problem, which could be viewed as a specific instance of ER problem. Our preliminary experiment of using a simplified version of RelDC framework for entity resolution has demonstrated the effectiveness of our approach. This paper is, however, only a first step in exploiting ER techniques for video data cleaning tasks. Our current approach has numerous assumptions and limitations: (1)The approach assumes that color of clothing is a strong identifier for a person on a given day; if several people wear similar color clothes and have similar activities, it is hard to distinguish them using the current approach. (2)If several people appear together, it is sometimes hard for the algorithm to correctly separate these subjects, and this negatively affects the result. Our future work will explore how additional features derived from video, as well as additional semantics in the form of context and metadata (e.g., knowledge of building layout, offices, meeting times, etc.) can be used to further improve person identification.

References

[1] M.-F. Balcan, A. Blum, P. P. Choi, J. Lafferty, B. Pantano, M. R. Rwebangira, and X. Zhu. Person identification in webcam images: An application of semi-supervised learning. In *In ICML Workshop on Learning from Partially Classified Training Data*, 2005.

[2] S. Chaudhuri, K. Ganjam, V. Ganti, R. Kapoor, V. Narasayya, and T. Vassilakis. Data cleaning in Microsoft SQL Server 2005. In *ACM SIGMOD Conference*, 2005.

[3] S. Chen, D. V. Kalashnikov, and S. Mehrotra. Adaptive graphical approach to entity resolution. In *Proc. of ACM IEEE Joint Conference on Digital Libraries (ACM IEEE JCDL 2007)*, Vancouver, British Columbia, Canada, June 17–23 2007.

[4] Z. Chen, D. V. Kalashnikov, and S. Mehrotra. Exploiting relationships for object consolidation. In *Proc. of International ACM SIGMOD Workshop on Information Quality in Information Systems (ACM IQIS 2005)*, Baltimore, MD, USA, June 17 2005.

[5] D. V. Kalashnikov and S. Mehrotra. Domain-independent data cleaning via analysis of entity-relationship graph. *ACM Transactions on Database Systems (ACM TODS)*, 31(2):716–767, June 2006.

[6] D. V. Kalashnikov, S. Mehrotra, S. Chen, R. Nuray, and N. Ashish. Disambiguation algorithm for people search on the web. In *Proc. of the IEEE 23rd International Conference on Data Engineering (IEEE ICDE 2007), short paper*, Istanbul, Turkey, April 16–20 2007.

[7] D. V. Kalashnikov, S. Mehrotra, and Z. Chen. Exploiting relationships for domain-independent data cleaning. In *SDM*, 2005.

[8] D. V. Kalashnikov, S. Mehrotra, and Z. Chen. Exploiting relationships for domain-independent data cleaning. In *SIAM International Conference on Data Mining (SIAM Data Mining 2005)*, Newport Beach, CA, USA, April 21–23 2005.

[9] A. McCallum and B. Wellner. Object consolidation by graph partitioning with a conditionally-trained distance metric. In *KDD Workshop on Data Cleaning, Record Linkage and Object Consolidation*, 2003.

[10] R. Nuray-Turan, D. V. Kalashnikov, and S. Mehrotra. Self-tuning in graph-based reference disambiguation. In *Proc. of the 12th International Conference on Database Systems for Advanced Applications (DASFAA 2007), Spinger LNCS*, Bangkok, Thailand, April 9–12 2007.

[11] R. Vaisenberg, S. Mehrotra, and D. Ramanan. Semartcam scheduler: *ema*ntics driven eal-ime data collection from indoor *am*era networks to maximize event detection. *J. Real-Time Image Processing*, 5(4), 2010.