Learning to Recognize Familiar Faces in the Real World

Lijin Aryananda

Abstract—We present an incremental and unsupervised face recognition system and evaluate it offline using data which were automatically collected by Mertz, a robotic platform embedded in real human environment. In an eight-day-long experiment, the robot autonomously detects, tracks, and segments face images during spontaneous interactions with over 500 passersby in public spaces and automatically generates a data set of over 100,000 face images. We describe and evaluate a novel face clustering algorithm using these data (without any manual processing) and also on an existing face recognition database. The face clustering algorithm yields good and robust performance despite the extremely noisy data segmented from the realistic and difficult public environment. In an incremental recognition scheme evaluation, the system is correct 74% of the time when it declares “I don’t know this person” and 75.1% of the time when it declares “I know this person, he/she is...” The latter accuracy improves to 83.8% if the system is allowed some learning curve delay in the beginning.

I. INTRODUCTION

Our target goal is a robot which can learn to recognize familiar individuals in a real world setting, motivated by the increasingly popular goal of integrating robots in the home. For human-centric tasks, the robots must be able to not only recognize each family member, but also to learn about various people’s roles in the household – who is the elderly person, who is the young child, etc. Among many technological challenges, we focus on two particular topics: unsupervised face recognition and operation in the real world. In this paper, we present and evaluate an incremental unsupervised face recognition system using data automatically acquired by Mertz, an interactive robot embedded in real human environment. Mertz was placed at public spaces for eight days (see figure 1) and has to perform the following tasks:

- operate continuously for several hours each day;
- attract passersby to approach the robot and engage them in spontaneous social interaction, e.g., by visual tracking and simple verbal exchanges;
- regulate the interaction to collect data from as many people as possible;
- detect, segment, store, and process face images and voice samples from each individual during interaction;
- use tracking and spatio-temporal assumptions to obtain a set of face sequences, where each sequence is assumed to contain faces of an individual;

We begin by describing the robot implementation, interaction results, and the automatically generated face data. We then present a novel face clustering algorithm and evaluate it offline using face sequences collected by the robot (without any manual processing) and an existing face database [1]. Lastly, we incorporate the clustering algorithm into an incremental unsupervised face recognition system and evaluate it offline using the same face data. If fully integrated into the robot, the system’s task is somewhat equivalent to sitting at a lobby, seeing hundreds of people in 8 days, and learning to recognize faces of 15 most encountered individuals.

II. RESEARCH FOCUS AND RELATED WORK

A. Operation in the Real World

Despite tremendous progress in robotics and face recognition research, the gap is still large between research prototypes and readily deployed applications. Long-term operation and robustness to a wide range of environments are still major challenges. We believe that chaining all subsystems together and interfacing them with real human environment is crucial for understanding the extent of these challenges. Mertz has to engage in spontaneous interaction with many passersby without any supervision for eight days (37 hours total) at 4 different locations at the MIT Stata Center lobby. This setup forces the robot to deal with drastic lighting and acoustical variations, simultaneous interaction with multiple people, and error propagation, from raw sensor input to face detection, tracking, and recognition.

The importance of robot reliability and robustness has been explicitly addressed by Graef and Bischoff [2]. Deployment in a public venue have been explored in museum
tour-guide robots [3], [4]. Some social robotic platforms have also been developed to operate for longer time scales outside the laboratory [5], [6], [7], [8], [9], [10]. These projects are very valuable in identifying rarely addressed limitations, e.g. speech recognition performance degradation in public environments [5], [7]. Individual recognition was explored in some of these projects, using a magnetic card-stripe reader in [5] and wireless RFID in [7].

B. Incremental and Unsupervised Face Recognition

Most of face recognition research concentrate on the supervised classification problem: given a set of manually labelled training data, find the correct person label for new test data [11]. In this project, we propose an unsupervised approach which would allow for incremental updates of the training set with more recent data and new individuals over time, starting from an empty database. This is aligned with the conclusion of a long-term human-robot social interaction study that robots should be able to “get to know” each frequently encountered individual [5] in order to establish long-term relationships.

Unsupervised face recognition has been explored for various purposes and applications [12], [13], [14], [15], [16]. Like ours, these systems (except for [16], [14]) also employ a video-based approach to face recognition, where each data sample is a sequence of faces instead of single face images. We later compare our experimental results with those of Raytchev and Murase [12].

III. FINDING, INTERACTING WITH, AND COLLECTING DATA FROM PEOPLE

Figure 2 illustrates the robot’s system architecture. The robot has to organize these subsystems to solicit spontaneous interaction and regulate these interactions to generate as much data as possible from people. We describe some relevant details here, but full system implementation is described in [17], [19]. Figure 3 shows a snapshot of the attention output while the robot interacted with two people simultaneously.

For obtaining face sequences, the robot combined the face detector [20], the KLT-based tracker [21], and some spatio-temporal assumptions to achieve same-person tracking. For every detected face, the robot activates the tracker for subsequent frames. All contiguous face images assumed to belong to a person (using segmentation initially produced by the face detector) are combined as a sequence. The robot then assigns a unique index for each sequence and stores it into the data set.

A. Summary of Observation

Evaluation and manual labelling of the collected data yield a number of observations below.

- The robot encountered at least 510 people, eighteen of which came by on multiple days.
- The robot interacted with at least one person about 30% of the time. About 16% of these active moments, the robot interacted with multiple people concurrently.
- Most people (97%) interacted continuously with the robot for 2 minutes, while a few people interacted for longer sessions, up to 14 minutes.

B. The Collected Face Data

In eight days, the robot autonomously collected 4,250 face sequences. The robot then automatically eliminated all sequences containing less than 8 images and processed the remaining. The final data set contains 2,025 face sequences formed by 134,252 face images. Figure 4 shows some of these sample face sequences. There are a lot of variations (orientation, facial expression, lighting, etc) within each face sequence, which is valuable for encoding maximal information about a person’s facial appearance, but challenging for recognition. We visually inspected 25% of the data set in three contiguous segments. On the bottom of figure 4 is the proportion of segmentation error, occlusion, and pose variations in this annotated set.

IV. UNSUPERVISED FACE SEQUENCE CLUSTERING

As described in the last section, the robot autonomously collected 2,025 face sequences. Each sequence is unlabelled and assumed to contain face images of one individual. The task of the clustering system is to cluster these sequences...
into classes such that each class corresponds to one individual. The number of individuals is unknown to the system. Previous studies have shown that the intraclass distance of a person’s face is sometimes larger than the interclass distance between two people’s faces [22]. Thus, the task of clustering face images is challenging and cannot rely on existing distance-based clustering methods. Moreover, many existing clustering algorithms require an a priori number of clusters, which is not available in our case.

Our clustering solution employ the following properties:

- Use of face sequence. Since the robot allows for situated face tracking, we use this freely available information to collect face sequences and employ a video-based instead of image-based face recognition approach.

- Use of local features. We are using SIFT (Scale Invariant Feature Transform) to describe the face sequences [23]. SIFT has been shown to provide robust nearest-neighbor matching despite scale changes, rotation, noise, and illumination variations. Its invariance capacity is crucial in allowing our system to deal with the high pose variations and noise.

- Sparse face alignment. This is an important design choice, as we want to step away from the requirement for a precise alignment, which has been shown to cause a large impact on face recognition performance [24].

- Clustering of local features. We represent each data sample (face sequence) as a set of SIFT keypoints extracted from different face parts. The task of clustering these local features in the 128-dimensional feature space is not straightforward. We develop a clustering algorithm for local features based on the simple intuition that if two sequences belong to the same individual, there will be multiple occurrences of similar local image patches between these two sequences. In addition to providing a sparse alignment, the face regions are also utilized to enforce geometrical constraints in the clustering step.

Figure 5 illustrates the steps of the face sequence clustering procedure, described in details below.

**A. Feature Processing**

A sequence of image $S_i$ is defined as $S_i = \{1m_{i,j} | j \in [1, L_i]\}$, where $L_i$ is the number of images in $S_i$. For each image $1m_i$ in each sequence $S_i$ (160x120 pixels), we use the Harris corner detector to identify a set of interest points [25], compute one SIFT 128-dimensional feature vector for each interest point (using fixed scale), and group the results in six batches based on which face region (R1-R6) the interest point is located in (see figure 5 left). The number of keypoints per face image ranges from 1000-3000 depending on image size. For each batch of keypoints produced by each face sequence, we perform k-means clustering to compute 50 keypoint prototypes [26].

Thus, as a final feature output, each sequence $S_i$ is mapped to $O_{i,1}, \ldots, O_{i,6}$, where $O_{i,m}$ is defined as $O_{i,m} = \{C_{i,m,p} | C_{i,m,p} \in \mathbb{R}^{128}; p = 1, \ldots, 50\}$, where each $C_{i,m,p}$ for $p = 1, \ldots, 50$ is one of the 50 prototype 128-dimensional vectors output.

**B. Sequence Matching**

We define the sequence matching function $\Psi(\cdot)$, which takes in two inputs:

- a sequence input $S_i$ which has been converted to six sets of 50 feature prototypes $C_{i,m,p}, m = 1, \ldots, 6, p = 1, \ldots, 50$.

- the rest of the data set $R = S_x \setminus [x \neq i, x \in [1, L_i]]$, which have been converted to $C_{x,m,p}, i = 1, \ldots, L, m = 1, \ldots, 6, p = 1, \ldots, 50$.

We describe each step of the sequence matching function $\Psi(\cdot)$ using the following pseudo-code:

```
SequenceMatching($C_{i,m,p}, R$)
FOR each m
    FOR each p
        Find K nearest neighbors to $C_{i,m,p}$ in region m in R
        Define $Count_i(x, m, p)$ to be the number of nearest neighbors that come from region m in sequence $S_x$
    ENDFOR
    Define $Count_i(x, m) = \sum_{p=1}^{50} Count_i(x, m, p)$
    Sort $Count_i(x, m)$ in descending order
    Take the top N elements in the sorted $Count_i(x, m)$
```

Fig. 5. The Unsupervised Face Sequence Clustering Procedure. For each sequence, we extracted SIFT features, group them based on the face regions shown on the left image, and compute prototypes. The sequence matching procedure compares each sequence’s feature set to the rest of the sequences in the data set and produces an output match set containing one or more sequences.
Sequence $S_x$ matches $S_i$ iff $\text{Count}_i(x, m)$ is in the top $N$ positions for all $m = 1, \ldots, 6$.

There is one missing detail, i.e. if the value of any element in the sorted $\text{Count}_i(x, m)$ list is $< C\%$ * the first element for some parameter $C$, this element and the rest of the elements in this sorted list of length $N$ are excluded.

C. Sequence Clustering

Using the sequence matching algorithm above, we compare each sequence $S_i$ against the rest of the data set $R = \{S_x|x \neq i, x, \ldots, L\}$, to produce an output set $M_i$ containing matches for $S_i$. If the output set $M_i$ is not empty, the system will combine $S_i$ and each element $S_x$ in $M_i$ into a cluster. This process is repeated for each $M_i$ from each sequence $S_i$. This clustering step is performed greedily such that if any two clusters contain matching elements, the two clusters will be merged together.

D. Evaluation Across Different Parameter Values and Data Set Sizes

As with any algorithms, it is crucial to assess robustness across different parameter values. The clustering algorithm involves 3 main parameters: N, K, C (see pseudo-code above). We perform a series of evaluation using various contiguous subsets of different sizes from the data set of 2025 face sequences, using different parameter values. Evaluation results indicate that for data sets containing 300 sequences or above, two of the three algorithm parameters (N and K) can be varied without much effect on the clustering performance. Essentially, we can use N=30 and K=10 for all cases, except when the data set contains 300 or more sequences. However, smaller data sets are not as immune to parameter changes. In particular, parameter N which is essentially correlated to the clustering aggressiveness, has to be kept low for good performance. This makes sense because the clustering algorithm is not purely distance based and thus benefits from having more data points. Detailed results and the corresponding parameter specification strategies are available in [17].

Figure 6 shows the accuracy of the sequence matching algorithm when performed on data sets of different sizes when the parameter C is varied. This accuracy measure is defined as the percentage of occasions when every element in the matching output set declared by the algorithm is indeed a correct match to the input sequence. Like the parameter N, C is also correlated to the clustering aggressiveness. These results indicate that the larger the data set is, the less impact large C value (more aggressive clustering) has on the clustering accuracy. Based on these results, we conclude a strategy, i.e. to linearly correlate parameter C to the data set size. Please note that although higher values of C clearly obtain the best results, keep in mind that a high accuracy value reflects that merging errors are low, but says nothing about the splitting errors.

E. Evaluation Using A Different Data Set

We also analyze the clustering performance using the Honda/UCSD Video Database [1], containing 72 video sequences from 20 individuals. Our downloaded version is missing the last person’s data, thus leaving only 19. Each sequence contains one person, rotating and moving his/her head in different order and speed. We half the video resolution to 160x120 to match our setup and apply the same exact processing.

With the most conservative parameter setting necessary for small data sets (N=3,K=10,C=70%), the clustering algorithm formed clusters correctly for 18 people. Ten of these clusters are perfect, i.e. contain all sequences that exist from the corresponding person. Eight of these contain 40-67% of existing sequences. One person’s sequences was split into two clusters. No cluster merging failure occurred.

F. Comparison to Related Work

For comparison, we formulate our results to match the performance metric used in [12], as shown in figure 7. The performance metric includes two error types: the number of mistakenly clustered sequences and the number of sequence in clusters with < 50% purity. Using this performance metric, we present our results using one data set of 2025 sequences and two sets of 500 sequences. For the latter, we display results using four different C parameter values, ranging from 0-70%, since we have observed that the smaller data set is more susceptible to error with a less conservative (lower) C parameter value. We again use the same values for N=30 and K=10. For both data sets, our results are slightly better, except for when C is reduced to 30% or less.
V. INCREMENTAL UNSUPERVISED FACE RECOGNITION

We incorporate the face clustering solution described above to implement an integrated system for unsupervised and incremental face recognition, as illustrated in figure 8.

![Incremental Unsupervised Face Recognition System](image1.png)

**Fig. 8.** The unsupervised and incremental face recognition system. It consists of two separate training sets, an unlabelled one for the clustering system and a labelled one for the recognition system. Both training sets are initially empty. Over time, the system incrementally builds a labelled training set using self-generated clusters and then uses this training set to recognize each sequence input.

- Initially, both training sets are empty.
- Each face sequence is fed into the system and simply stored in the clustering training set.
- When the clustering training set contains 300 sequences, a batch clustering is performed. The 15 largest resulting clusters are then transferred to the recognition training set where each cluster corresponds to a labeled class.
- Now that the supervised recognition system is ready, each image from the face sequence input is also passed to the supervised recognition system one at a time. The supervised recognition system is implemented using an image-based variant of the face sequence matching algorithm, i.e. the input is six sets of 50 feature prototypes extracted from only one image, instead of a sequence of images. The system produces a final hypothesis after a minimum number of face images. The sequence input is declared to belong to a particular person in the recognition training set as soon as \( \geq 50\% \) recognition rate is obtained. Otherwise, it is declared to be an unknown person.
- The sequence input is also passed to the clustering system, which either integrates it into an existing cluster, uses it to form a new cluster, or leaves it as a singleton. This incremental change is subsequently reflected by transferring the 15 largest clusters to the recognition training set, and the loop continues.

A. The Incremental Recognition Performance

**Fig. 9.** The incremental recognition results as the system receives each sequence input one at a time. The lower sub-plot shows the number of sequences in the labeled training set. The upper sub-plot shows the accumulated number of correct and incorrect hypothesis when the system "knows" the person and who he/she is, which is correct 75.1% of the time. After some learning curve, the accuracy improves to 83.8%. The middle sub-plot shows that the "I don't know this person" hypothesis is correct 74% of the time.

![Incremental Recognition Results](image2.png)

**Figure 9** shows the results of the process described above. The lower sub-plot shows the number of sequences in the incrementally constructed labelled training set, which increases over time.

When the system decides that "I don’t know this person", the middle sub-plot shows that it is correct 74% of the time. When the system hypothesizes that "I know this person, he/she is ...", the upper sub-plot shows that it is correct 75.1% of the time. The slope of the number of incorrect hypotheses decreases over time as the size of the labelled training set increases. If we calculate the recognition performance after some delay, as shown by the blue and red dotted lines, the performance improves to being correct 83.8% of the time.

We then take a deeper look of the recognition performance for some familiar individuals who encountered the robot on multiple days. Figure 10 shows two sample clusters of a woman and a man who came to interact with the robot on seven and two different days respectively and two sample of bad clusters.

B. The Incremental Clustering Accuracy

The incremental recognition performance above is certainly dependent on the clustering accuracy. Figure 11 shows the incremental clustering results when using 6 different correlation functions between the parameter C and data set size.

The clustering system for the incremental recognition test above was implemented using correlation function \( inc3 \). From 291 people with multiple sequences in the data set, the clustering algorithm formed 151 clusters correctly. Half of these are perfect, i.e. contain all existing sequences belonging to the corresponding individual. The algorithm made 22 merging failures, with the largest merged cluster containing 3 individuals. There were 26 cluster with splitting failures, i.e. contain sequences from the same individuals. Figure ?? top illustrate a bad cluster sample where multiple individuals were falsely merged together, most likely because of the dominant background region. Figure ?? bottom shows another bad cluster consisting of only background images which were initially falsely detected as faces.

Across the six different correlation functions, the clustering system is generally capable of generating clusters with low errors for roughly half of the individuals in the data...
clustering algorithm generated stable performance across a wide range of parameter settings.

REFERENCES


