PEOPLE RE-DETECTION USING ADABOOST WITH SIFT AND COLOR CORRELOGRAM

Lei Hu\textsuperscript{1,2,3}, Shuqiang Jiang\textsuperscript{1,2}, Qingming Huang\textsuperscript{2,3}, Wen Gao\textsuperscript{2,4}

\textsuperscript{1}Key Lab of Intelligent Information Processing, Chinese Academy of Sciences, Beijing 100080, China
\textsuperscript{2}Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China
\textsuperscript{3}Graduate University of Chinese Academy of Sciences, Beijing 100049, China
\textsuperscript{4}Institute of Digital Media, Peking University, Beijing 100871, China

\{lhu, sqjiang, qmhuang, wgao\}@jdl.ac.cn

ABSTRACT

People re-detection aims at performing re-identification of people who leave the scene and reappear after some time. This is an important problem especially in video surveillance scenarios. In this paper, we present a method of people re-detection within the context of visual sequence in single-camera setup. We consider re-detection as a binary classification problem, where both global and local descriptors are employed for training strong classifier on-line with Adaboost to distinguish a newly detected people as tracked or new occurrence. The strong classifier will be updated while match is ascertained. A predetermined classifier with well-chosen threshold is employed as assistant of training examples collection. We test the performance of our approach on 4 different scenes including 51 video sequences taken from the CAVIAR database and 4 video sequences shot by ourselves. The results show that our re-detection algorithm can robustly handle variations in illumination, pose, scale, and camera-view.

Index Terms— people re-detection, SIFT, Color Autocorrelogram, Adaboost

1. INTRODUCTION

In Visual Surveillance, re-occurrences of a same person normally happen. People re-detection aims at finding re-occurrences of detected or tracked people on a cursory level within a not-long period of time, for example, one day. Figure 1 shows a scenario which people re-detection focuses on. Therefore, this problem is based on the assumption that people undergo only minor changes between their all occurrences in a period, for instance, that a person wears the same clothes in all scenes. It holds for most situations, which is our focus in this work.

There are a number of applications in single camera setup that benefit from the information that is extracted by people re-detection, including the following. 1) Long period behavior analysis: Behavior analysis is always one of the ultimate targets in visual surveillance. Most of these works before are based on short time snapshots of person which could not provide enough information for completely extracting the behavior patterns of people. Assume that we can successfully identify a same person in discrete time space, we will be able to gather enough data for behavior analysis. 2) Connecting interrupted tracking: People re-detection can be used to improve the tracking result by connecting parts of the person’s trajectory, which have been split due to occlusion, shot boundaries or other distortions.

Besides this, the information of these re-occurrences usually conveys an important part of visual surveillance in multi-camera setup. Surveillance of wide areas requires a network of cameras. It is not always possible to have overlapping camera views in this case. The observations of the same person can be widely separated in time and space in such a scenario. Therefore, how to judge whether the person observed in one camera and the one viewed in another camera are the same person is exactly a people re-detection problem.

Person re-detection is very complex, since persons are articulated, move arbitrarily, and often wear multi-colored dress. Javed et al. [9] use various features based on space-time (entry/exit locations, velocity, travel time) and appearance (color histogram). A probabilistic framework is developed to detect best matches. Bird et al. [4] detect loitering individuals by matching pedestrians intermittently spotted in the camera field of view over a long time. Snapshots of pedestrians are extracted and divided into thin horizontal slices. The feature vector is based on color in each slice and Linear Discriminant Analysis is used to reduce the dimension.

In recent years methods of combining local features and color features [2][6][15] have become most popular because they have solved a number of object recognition problems. Patwardhan et al. [13] presented a novel scheme to generate a graphical model of the appearance and relationship between objects. Each segment of person is modeled using the combination of the color histogram and the SIFT [10] features. Schügerl et al. [14] proposed special applications of people re-detection in multimedia content analysis, retrieval, and authoring. A voting framework is developed to combine SIFT and MPEG-7 Color Descriptors [11] for detecting re-occurrences of

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{image.png}
\caption{A scenario which people re-detection aims at. In the left image(a), a person appears in this scene, and then moves out of view in the middle image(b). After a period of time, the same person income again, which is called re-occurrence.}
\end{figure}
people in image and video data sets.


This paper presents a method of people re-detection within the context of visual sequence in single-camera setup. We consider re-detection as a binary classification problem, where both global and local descriptors are employed for training on-line weak classifiers to distinguish a new detected people as tracked or new occurrence.

At the very beginning we employ a direct combination of selected features with well-chosen threshold for assisting training examples collection. The weak classifiers are combined into a strong classifier using Adaboost. Then the strong classifier is used to classify whether the new appearing person has been tracked, and will be updated while matching is ascertained. Our contributions in this paper are: 1) employing both global and global features on people re-detection; 2) a new training examples collection scheme; 3) a novel strong classifier update framework specified for this problem.

The rest of this paper is organized as follows: The descriptor selection and similarity measures are detailed in Section 2. In Section 3, we introduce a novel framework of training weak classifiers ensemble with Adaboost for people re-detection assisted by a predetermined classifier. The experiment results are provided in Section 4. Section 5 is our conclusion.

2. DESCRIPTOR SELECTION AND SIMILARITY MEASURES

2.1. Descriptor Selection

In order to develop a people re-detection system with great performance we have to select appropriate global and local features.

A large number of color descriptors for images and image regions have been proposed over the last 15 years. In [8], Huang et al. present a high-efficient Color Autocorrelogram which is widely used on image retrieval and matching. A Color Autocorrelogram expresses how the spatial correlation of pairs of colors changes with distance which could show great performance on object re-detection, therefore, we choose Color Autocorrelogram as the global feature in our method.

Extensive studies [12] have shown that SIFT descriptors outperform other local descriptors for object recognition on various types of image data, including 3D objects and real world scenes. The SIFT descriptor is based on a multidimensional gradient histogram and has been proposed in [10]. Due to its good invariance to illumination changes as well as perspective changes, this texture descriptor has been successfully used in various works.

2.2. Similarity Measures

The similarity $\tilde{S}(A \rightarrow B)$ of person A in the example frame to a person B in the queried frame is computed as a product of the color and feature similarities:

$$
\tilde{S}(A \rightarrow B) = \frac{D(A \rightarrow B)}{D(A, B)}
$$

where, $D(A, B)$ is the a normalization of relative distance measure [8] between autocorrelograms of A and B, and $D(A \rightarrow B)$ is the SIFT keypoints feature similarity:

$$
D(A, B) = \frac{1}{m \times d} \sum_{i \in [m], k \in [d]} 2 \times \frac{\gamma_{c_i,c_j}(A) - \gamma_{c_i,c_j}(B)}{1 + \gamma_{c_i,c_j}(A) + \gamma_{c_i,c_j}(B)}
$$

where, $\gamma_{c_i,c_j}(A)$ is the squared Euclidean distances between the $c_i$th and $c_j$th keypoints in A.

The similarity $\tilde{S}(A \rightarrow B)$ of person A in the example frame to a person B in the queried frame is computed as a product of the color and feature similarities:

$$
\tilde{S}(A \rightarrow B) = \frac{1}{N_A} \sum_{i=1}^{N_A} e^{-d(S^i_A, S^j_B)}
$$

where, $i = \arg \min_j d(S^i_A, S^j_B), j = 1, 2, \ldots, N_B$

### Table 1. The variable definitions of equations above

| $\gamma(\cdot)$ | auto correlogram matrix of person |
| $m$ | the colorspace are quantized into m colors $c_1, \ldots, c_m$, for instance, in RGB colorspace each color are divide into 4 bins, so we have $4 \times 4 \times 4 = 64$ color bins in our histogram |
| $b$ | the number of distance set, we use $1, 3, 5, 7$ in our experiments |
| $c_i$ | color features $i$ of autocorrelogram, $i = 1, \ldots, m$ |
| $N_A$ | the number of keypoints in A |
| $S^i_A$ | the $i$th keypoint in A |
| $d(\cdot, \cdot)$ | the squared Euclidean distances between the $j$th keypoint in B with the $i$th keypoint in A |
| $\tilde{\gamma}$ | the index between the keypoints in B are the best matches for $i$th keypoint in A |

3. PEOPLE RE-DETECTION USING ADABOOST

People re-detection using Adaboost works by constantly updating a collection of weak classifiers to separate the reappearance of tracked person from the new entering object. The weak classifiers can be added or removed at any time to reflect changes in the tracked person’s appearance or incorporate new information derived from new detected incomer.

Each weak classifier is trained on positive and negative examples where, by convention, we term examples coming from the appearance difference between several occurrences of a same person in a period of time as positive examples and examples coming from the difference between different persons as negative examples. The strong classifier, calculated using Adaboost, is then used to classify the new appearing person as tracked person or fresh incomer. Once the identification for the current incomer is completed, we add it into the training examples set as a new negative or positive difference example and then train a new weak classifier which will be added to the ensemble, and repeat the process all over again.

3.1. The weak classifier

Let $\{D_i, i = 1, \ldots, N\}$ be the training difference example set where $N$ is the number of the training examples. Every example is associated with a label $\{l_i, i = 1, \ldots, N\}$ and $l_i = 1$ if the incomer snapshot is ascertained as extracted from one person tracked, otherwise $l_i = 0$. Each difference example is represented as a d-dimensional feature vector that consists of some global and local information.

The weights, $\frac{1}{N_p}, \frac{1}{N_n}$, are initially set to positive training samples and negative training samples respectively where $N_p$ is number of positive sample and $N_n$ is number of negative sample.

The general form of an Adaboost weak classifier discriminant function is defined as:

$$
h(D_i) = \begin{cases} 
1, & h^T \cdot D_i \geq 0 \\
-1, & \text{otherwise}
\end{cases}
$$

where $h$ represents the discriminating hyper-plane trained in a least-squares manner.
Algorithm 1: Strong Feature Classifier Ensemble

| Input | persons entering sequence \( \{P_1, \ldots, P_m\} \) 
|-------|-------------------------------|
| Output| People re-detection result sequence: \( \{r_1, \ldots, r_n\} \) 

Initialization (Repeat in 1...):

1. Collecting data set \( \{D_i\}_{i=1}^N, \{l_i\}_{i=1}^N \) as Algorithm 2
2. Initialize weights \( \{w_i\}_{i=1}^N \) to be \( \frac{1}{\sqrt{N}}, \frac{1}{\sqrt{N}} \) respectively.
3. For \( t = 1, \ldots, T \),
   (a) Train weak classifier \( h_t \)
   (b) Set \( \text{err} = \sum_{i=1}^N w_i |h_t(D_i) - l_i| \)
   (c) Set weak classifier weight \( \alpha_i = \frac{1}{2} \log \frac{1-\text{err}}{\text{err}} \)
   (d) Update example weights \( w_i = w_i e^{\alpha_i (h_t(D_i) - l_i)} \)
4. The strong classifier is given by \( \text{sign}(H(x)) \) where \( H(x) = \sum_{i=1}^T \alpha_i h_t(x) \)
5. Repeat until \( H_t \) perform better than predetermined classifier

For each newly entered person \( P_j \) do:

1. Extract \( \{D_i\}_{i=1}^N \) examples
2. Test the examples using the strong classifier \( H(x) \), label as \( \{l_i\}_{i=1}^N \)
3. Output \( r_j \in [0, \ldots, j - 1], 0 \) present no match.
4. Remove \( K \) oldest weak classifiers
5. Revise weights \( w_i = w_i \left( 1 - \frac{N_i}{\sum_{k=1}^{j-1} N_k} \right), i \in [1, \sum_{k=1}^{j-1} N_k] \)
6. Initialize weights \( w_i = \frac{1}{\sum_{k=1}^{j-1} N_k} \) for new examples
7. For \( l = K + 1, \ldots, T \), (Update Weights)
   (a) Choose \( h_t(x) \) with minimal error \( \text{err} \)
   (b) update \( \alpha_t \) and \( \{w_i\} \), then Remove \( h_t(x) \) from \( \{h_{K+1}(x), \ldots, h_T(x)\} \)
8. For \( t = 1, \ldots, K \), (Add new weak classifiers)
   (a) Train weak classifier \( h_t \)
   (b) Compute \( \text{err} \) and \( \alpha_t \), Update example weights \( \{w_i\} \)
9. The updated strong classifier is given by \( \text{sign}(H(x)) \) where \( H(x) = \sum_{t=1}^T \alpha_t h_t(x) \)

The temporal coherence of appeared persons is exploited by maintaining a list of \( T \) classifiers that are trained over time. Once an incomer is detected and then be judged as a tracked person or a new incomer, we discard the oldest weak classifier, train a new weak classifier on the data added newly available examples, and reconstruct the strong weak classifier. That means a person does not reappear in a period of time will be disregarded as a tracked person.

Specific person to be tracked can be incorporated into the training data set as one or more examples that participate in the training of weak classifier, but cannot be removed in the update stage.

The specific algorithm we use is given in Algorithm 1.

3.2. Training data set collection

Appropriate and sufficient training data set is important to the training of weak classifiers. Certainly, predetermined person can be incorporated into the data set, however, it is hard to predict all the people who will occur in the scene. Therefore, we should collect training examples directly from the video sequence.

We present a novel framework in assisting the collection of training examples at the very beginning of people re-detection process. A predetermined classifier which has been trained on some examples off-line is employed as the initializing assistant. This classifier \( C \) is actually a similarity measure which have been noted in section 2.2 with a well-chosen threshold \( \tau \):

\[
C(A \rightarrow B) = \text{sign}(\delta(A \rightarrow B) - \tau)
\] (6)

This classifier is used to generate labels on extracted training examples, until the strong classifier outperforms it. A general algorithm is given in Algorithm 2. Along with the number of training data increasing, old examples will be abandoned, as well as the examples extracted from persons who scarcely appear again.

Algorithm 2: Training Data Set Collection

| Input | entering person \( P^m \) 
|-------|-----------------------------|
| m     | tracked persons \( \{P_1^t, \ldots, P_m^t\} \) 
| s     | tracking persons \( \{P_1^s, \ldots, P_s^s\} \) 
|       | predetermined classifier \( C(x) \) 

Output: Given training examples \( \{(D_1, l_1), \ldots, (D_n, l_n)\} \) where \( l_i = 0, 1 \), for negative and positive examples respectively
1. Extract \( k \) snapshots from \( P^m \)
2. Compute all distance between the \( k \) snapshots, label with 1
3. For \( j = 1, \ldots, s \)
   Compute distance between snapshots of \( P^m \) and \( P_j^t \), label with 0
4. For \( j = 1, \ldots, m \)
   (a) Compute distance between snapshots of \( P^m \) and \( P_j^t \)
   (b) If \( \text{sign}(\sum_{a,b \in [k]} C(T_{a,b}^t \rightarrow T_{a,b}^m)) = 1 \), label with 0; if not, label with 1
5. Add the new examples into training data set

3.3. Strong feature classifier ensemble update

In the update state, the algorithm removes \( K \) old weak classifiers to make room for \( K \) new weak classifiers. However, before adding new weak classifiers one needs to update the weight of the remaining weak classifiers. Step (7) of Algorithm 1 in the update state updates the weights of the remaining weak classifier. Instead of training a new weak classifier on the whole training data set, the weak learner simply give the old examples a new distribution based on their origin weights, so that they need not be computed again, as step(5). This saves training time and creates a strong classifier as well as a sample distribution that can be used for training the new weak classifier, as is done in step (8).

4. EXPERIMENT

We implement the proposed method in C and test the performance on 4 video data sets including 51 video sequences taken from the CAVIAR[1] database and 10 video sequences shot by ourselves, which are divided by their different scenes. The video sequences including 86 persons in view and 320 times in total reappearing situation cover most of the simple and complicated scenarios in people re-detection.
No parameters are changed from one experiment to the next and in all cases the predetermined classifier are provided. In all cases we use a 512D feature vector per example extracted from snapshots of two person that consists of a 256-bin color autocorrelogram $L_1$, distance vector as well as the 256 smallest Euclidean distance between SIFT keypoints of two snapshots. The threshold of predetermined classifier is selected as 0.7.

In Figure 2, new occurrences of example persons in completely different camera views are compared to each other and are ascertained as same persons. The results show that our re-detection algorithm can robustly handle variations in illumination, pose, scale, and camera-view. The performances of our approach are shown in table 2. Results of people re-detection in corridor scene are fairly lower than that on other scenarios, because in the corridor video data there are only a few re-occurrences comparing to the number of appeared persons which causes much less positive training examples and greater computation loads. This demonstrates that our method are more effective on scenarios in which re-occurrences regularly happen. Another scene from which we acquired slightly lower accuracy is playground, too many distortion of human body while playing football games can explain it.

![Corridor: (a) frame 96  (b) frame 300  (c) frame 1440](image1)

![Playground: (a) frame 261  (b) frame 408  (c) frame 824](image2)

**Fig. 2.** Results of people re-detection. In the top row of indoor setup, a person goes into the shop and gets out of view at frame 300, then leaves the shop for the scene again at frame 1440. The green rectangle demonstrates the result of re-detection: the incomer matches a tracked person. The bottom row is a test on a video shot in a mini football game, and shows a similar result.

**Table 2.** The performance of our method on all test scenes

<table>
<thead>
<tr>
<th>Scene</th>
<th>people number</th>
<th>reoccur number</th>
<th>accuracy(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hall</td>
<td>12</td>
<td>80</td>
<td>89.3%</td>
</tr>
<tr>
<td>Corridor</td>
<td>58</td>
<td>38</td>
<td>70.4%</td>
</tr>
<tr>
<td>Road</td>
<td>10</td>
<td>20</td>
<td>95%</td>
</tr>
<tr>
<td>Playground</td>
<td>6</td>
<td>106</td>
<td>85.9%</td>
</tr>
</tbody>
</table>

### 5. CONCLUSION

We consider re-detection as a binary classification problem. An ensemble of weak classifiers is trained and updated on-line to distinguish between features of new detected people as tracked or new occurrence. We select color autocorrelogram and SIFT as global and local features, respectively. A predetermined classifier with well-chosen threshold are employed as assistant of initialization. The effectiveness of this method was demonstrated with several video sequence data sets. We plan to further enhance the capability of our approach by applying dimension reduction which could prune the extraneous information on our selected features.

### ACKNOWLEDGEMENT

This work was supported by "Science100 Plan" of Chinese Academy of Sciences under Grant 99T3002T03.

### 6. REFERENCES


1351