Pedestrian Re-identification by Graph Clustering

Hai-Sheng CHEN¹,a, Dan TU²,b,* , Jun LEI³,c
College of Information System and Management,
National University of Defense Technology, Changsha, Hunan, China
1538767813@189.cn
*Corresponding author

Keywords: Pedestrian, Graph Partitioning, Multi-camera.

Abstract. Matching people in multi-camera views, known as pedestrian re-identification problem, is a challenge task. Searching a designated pedestrian at the entire monitoring scene has been achieved in previous work. However, the existing results only return a sequence of pedestrian images ranked by the similarity with the input image, rather than matched images. In this paper, we use graph partitioning methods to solve the pedestrian re-identification problem. We first get a similarity matrix by calculating the similarity between each image and others. Then we consider the matrix as an undirected graph and use graph partitioning methods to partition it. The result of graph partitioning corresponds to the classification of pedestrian images. The main contributions of this paper include 1) we estimate the number of pedestrians on the multi-camera views, 2) we label a same object ID for sample images of the same pedestrian.

Introduction

We capture pedestrian sample images through the pedestrian detection technology from the camera surveillance network, and achieve the aim of automatic classification for these sample images, labeling a same object ID for sample images of the same pedestrian.

In the previous work [1], researchers consider a pedestrian image as the probe image, and then search images which are more similar with the probe image according to the similarity index (the images of the same person are usually more similar). At last, they rank the results and return the first $t$ images. However, this kind of methods, especially the selection of the threshold $t$, neither guarantee to return all the captured images belonging to the selected pedestrian, nor guarantee that all the returned images belong to the probe pedestrian. In this paper, we return all the captured images for an arbitrary pedestrian though the number of captured images of each pedestrian is unknown and not equal, leading to the threshold $t$ useless.

We consider the person re-identification task as a clustering problem instead of searching and ranking. Specifically, in a surveillance network consisting of $n$ cameras, each camera captures $N_i$ pedestrian images, denoting $N = sum(N_1, N_2, \ldots N_n)$ as the number of all the captured sample images, and suppose $m$ pedestrians appear in the images. According to [2], we know those images from the same pedestrian are more similar in many features performance such as color, texture et al. Therefore, we classify the $N$ pedestrian images into $m$ clusters $(p_1, p_2, \ldots, p_m)$ according to similarities, and each cluster corresponds to each person. Then, we get the sequence of appearances for each pedestrian in the camera monitoring scene, and each sequence includes a person’s images captured at different time and place by an arbitrarily camera.

Different from the traditional methods which extract features of samples in order to directly cluster these samples by the distance between each other, we get a similarity matrix by calculating the similarities between each image and others, and then use graph partitioning to solve the clustering problem. The benefit of our method is that the size of the similarity matrix only depends on the number of samples, avoiding repeated similarity calculation in the iterative process, especially when the feature representation structure of samples is complex or the feature dimension
is high. In fact, dColorSIFT [1] describes the feature of a pedestrian image with a 10^6 dimensional vector, which products expensive computation cost.

The optimum solution for the graph clustering problem is a NP hard problem, and the different relaxation conditions correspond to several different common graph partitioning methods: SymNMF [3], a method based on the non-negative decomposition of binary graph; Spectral clustering method (SC) [4 5], consider the orthogonality but ignore non negative and discrete of the final solution, and is same with SymNMF when consider strictly orthogonality constraints; CAC [6], consistent approximate graph clustering (CAC), keep strictly non negative and effectively orthogonality.

All of the above methods need to set the clusters number before starting graph partitioning. However, the number of clustering is unknown in the real circumstance. Considering the modularity concept proposed in GN algorithm [7]—a classical algorithm for the network community partition problem, we consider the pedestrian similarity graph partitioning problem as a kind of network community division problem, and estimate the cluster number by using the modularity curve. In the community division problem, a community represents a person, and the number of different communities corresponds to different modularity value. Usually, the max value of modularity corresponds to the most likely number of communities.

![Image](image1.png)

**Figure 1.** The whole process of pedestrian re-identification.

The whole process of the pedestrian re-identification task is showed in Fig.1. Firstly, we randomly select some pedestrians’ images from the ETHZ dataset [8] to construct similarity graph, and then use three graph partitioning methods to solve the graph clustering problem. We use the clustering accuracy ($A_C$) and the standard mutual information ($MI$) to evaluate the clustering results [9]. All the three methods have a good performance in the experiment.

The rest of this paper is organized as follows. Section 2 introduces the feature representation of pedestrian images and similarity graph construction. In Section 3, we introduce several typical graph partitioning methods. Experimental results are presented in Section 4. Finally, conclusions are drawn in Section 5.

**Graph Construction**

When we cope with the pedestrian clustering problem in the multi-video surveillance scene, we first extract features of pedestrian images, and then construct the similarity graph.

**Features Representation of Pedestrian Samples**

Features representation of images is challenging in multi-camera pedestrian re-identification due to the large visual appearance changes caused by variations in view angle, lighting, background clutter and occlusion. In this work, we choose features which are more stable to re-identify the person who have multiple images. In the task of pedestrian searching, dColorSIFT [1] has achieved a good performance. Therefore, we also adopt dColorSIFT feature to solve our task.
In the experiment, all the pedestrian images are normalized to $32 \times 64$, and patches of size $10 \times 10$ pixels are sampled on a dense grid with a grid step size 4; 32-bin color histograms are computed in L, A, B channels respectively, and in each channel, 3 levels of downsampling are used with scaling factors 0.5, 0.75 and 1; SIFT features are also extracted in 3 color channels and thus produces a $128 \times 3$ feature vector for each patch. In a summary, each patch is finally represented by a discriminative descriptor vector with length $32 \times 3 \times 3 + 128 \times 3 = 672$. We denote the combined feature vector as dColorSIFT.

Graph Construction

Rui Zhao et al. [1] design an ingenious function to measure similarity, and we use this similarity function to calculate the similarity matrix. Then we use $p$-nearest neighbor method to convert the similarity matrix to $p$-nearest neighbor graph [10]. In the $N \times N$ similarity matrix, $w(i, j)$ represents the similarity between the image $i$ and image $j$. $w(i, j)$ is retained if image vertex $j$ is among the $p$-nearest neighbors of vertex $i$ and we can adopt the symmetrisation process to obtain a symmetric graph. Retain or discard each $w(i, j)$ as follows:

$$w(i, j) = \begin{cases} w(i, j), & \text{rank}(w(i, j)) \leq P \\ 0, & \text{otherwise} \end{cases}.$$

(1)

Where $\text{rank}(w(i, j))$ is the location of $w(i, j)$ in the rank sequence of all the pairwise similarities between image $i$ and others.

Graph Partitioning

The graph clustering problem is usually formalized as the graph cut or partitioning problem [15]. The common partition criteria include Minimum cut [16], RadioCut [12], Average cut [17], normalized cut (NCut) [11] and so on. These criteria will lead to discrete optimization problems, finding the optimal solution of the graph cut is NP-hard. Some relax conditions are considered when solving this problem and lead to several typical graph partitioning methods, and we use the methods to partition the graph respectively in our experiments.

Several Graph Partitioning Methods Comparison

Methods Comparison. One of the famous relaxations is discarding the discreteness condition and instead allowing solution vector to take arbitrary values in $\mathbb{R}$, which leads to spectral clustering (SC) methods [5, 12]. SC uses eigenvectors of the adjacency matrix derived from the data for partitioning of graphs, which is a well-established framework for graph clustering. In fact, SC considers the orthogonality but ignores the non-negativity and the discreteness of the final solution.

Therefore, the ideal relaxation should consider the non-negative constraints, the orthogonality constraints and the discreteness constraints. As a result, non-negative aplacian embedding [21] and Sym non-negative matrix factorization (NMF) [19, 20] (SymNMF is equal to SC under strict orthogonality constraints) can retain the non-negativity rigorously and enforce the orthogonality approximately at the same time.

A novel method named congruent approximate graph clustering (CAC) is proposed. CAC can retain the non-negativity rigorously and can enforce the orthogonality efficiently. In addition, it is closer to the optimal solution.

Measure Index. We use clustering accuracy ($AC$) and normalized mutual information ($\overline{MI}$) measure the clustering results [23]. $AC$ is defined as:

$$AC = \frac{\sum_{i=1}^{N} \delta(l_i, l'_i)}{N}.$$

(2)
\( \delta(x,y) \) is a delta function, the value is 1 when \( x = y \), otherwise is 0. \( l_i \) is the true class label of \( x_i \), \( l'_i \) is the assigned label by graph clustering methods.

The MI metric of two sets of clusters \( C \) and \( C' \) is defined as follows

\[
MI(C, C') = \sum_{c_i \in C, c_j \in C'} p(c_i, c_j') \log \frac{p(c_i, c_j')}{p(c_i)p(c_j')}
\]  
(3)

where \( p(c_i) \) and \( p(c_j') \) are the probabilities that a sample belongs to the cluster \( c_i \) and \( c_j' \), respectively, and \( p(c_i, c_j') \) is the joint probability. The normalized \( \overline{MI} \) as follows

\[
\overline{MI}(C, C') = \frac{MI(C, C')}{\max(H(C), H(C'))}.
\]  
(4)

where \( H(C) \) and \( H(C') \) are the entropies of \( C \) and \( C' \).

**Estimate the Number of Clusters**

The number of pedestrians is required when we use the graph partitioning methods mentioned in section 3.1.1. However, the number of pedestrians is unknown in the pedestrian similarity graph. So we need to estimate the number of pedestrian according to the similarity graph. In this work, we use the modularity concept proposed in GN algorithm to make the estimation.

GN algorithm [24] is a classical algorithm for Network community Division problem. The concept of betweenness is used in graph partitioning, the betweenness is defined: number of shortest paths from all vertices to all others that pass through that edge. The GN algorithm is as follows: remove the edges with the maximum betweenness from the network iteratively until the network is decomposed into each leaf node. In order to find the most likely number of communities the network conclude, modularity is proposed.

Modularity is defined as follows: If the network is divided into \( n \) groups, then we define \( n \times n \) symmetric matrix \( e \), the element \( e_{ij} \) represents the ratio between the number of edges linking nodes in community \( i \) and \( j \) and the number of all the edges in the whole network. \( T_{re} = \sum_i e_{ii} \) represents the ratio between the number of edges in the internal community and the number of all the edges in the whole network. The sum of rows(or cols) \( a_i = \sum_j e_{ij} \) represents the ration between the number of edges in the community \( i \) and the number of all the edges in the network. According to the definitions, we can get \( e_{ij} = a_i \times a_j \). So modularity function is defined as:

\[
Q = \sum_i (e_{ii} - a_i^2) = T_{re} - \|e^2\|.
\]  
(5)

\( \|e^2\| \) is the matrix norm of \( e^2 \), namely, the sum of all elements in \( e^2 \).

We consider the graph partitioning problem as a kind of Network community Division problem, where each community represents a pedestrian. With the number of communities increasing, the modularity value increases at first and then decrease after the peak. Therefore, we draw the modularity curve according to the similarity graph and consider the community number corresponding to the maximum value in the modularity curve as the estimate number of pedestrians.

**Experiment**

We use a part of ETHZ dataset [25] to conduct experiments, because the ETHZ is too large. In this section, we randomly select pedestrians’ images from the ETHZ dataset for experiments, and our experiments are based on 3 sub-datasets:
Dataset1: 146 sample images from 146 pedestrians, and we increase the number of samples of each pedestrian to get an expanded version, the larger one consists of 1440 images.

Dataset2: 146 sample images from 11 pedestrians, and the number of samples of each pedestrian is not exactly same but the difference is not large.

Dataset3: 837 sample images from 11 pedestrians, the number of samples of each pedestrian is not the same and the difference is large. The 11 pedestrians is the same with the last one, and the bigger number of one pedestrian reach 182, 353 while the smaller as 7, 10.

For each selected dataset, we get the similarity matrix by calculating the similarities between each image and others, and then use $p$-nearest neighbor method to construct a graph. Denoting $\text{sparse}_p$ as the value of $p$, we derive multiple graphs from similarity matrix by using different $\text{sparse}_p$. Then we partition these graphs and evaluate the results by $AC$ and $MI$.

Visualize the Graph

We get the similarity matrix by calculating the pairwise similarities of those pedestrians' images, then we construct the $p$-nearest neighbor graph based on the matrix of different dataset.

In the undirected graph, each vertex represents an image, and the weight of edge represents the similarity. In order to represent the result more clearly, we put gravitation on similarity, which means the distance of nodes having greater similarity is closer, and vice versa. The visual layouts of these graphs are presented in Fig. 2.

![Figure 2](image-url)

Figure 2. (a) the visual layout of 146 images from 11 pedestrians (b) the visual layout of 837 images from 11 pedestrians (c) the visual layout of 146 images from 146 pedestrians (d) the visual layout of 1440 images from 146 pedestrians.

We can see the images describing the same pedestrian show a good aggregation property as presented in Fig. 2a and Fig. 2b. While the images from different pedestrian distribute messily as presented in Fig. 2c and show the aggregation property around the pedestrian when the images number of each pedestrian increase as presented in Fig. 2d.

Evaluate the Clusters

In terms of Dataset2, we derive multiple graphs from similarity matrix by using different values of $\text{sparse}_p$, and then GN algorithm is used to compute the modularity curves which are showed in Fig. 3.
Denote max_modulity as the maximum value of the modulity, and num_community as the corresponding number of communities. The results are showed in Table 1. The optimal number of communities got by the modulity curve is related to the \( \text{sparse}_p \) value. When \( 7 < \text{sparse}_p \leq 25 \), the optimal number of communities is around 11.

Table 1. The maximum modulity value and the most probable communities number of each graph.

<table>
<thead>
<tr>
<th>( \text{sparse}_p )</th>
<th>5</th>
<th>7</th>
<th>8</th>
<th>10</th>
<th>12</th>
<th>15</th>
<th>20</th>
<th>25</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>max_modulity</td>
<td>0.8395</td>
<td>0.8440</td>
<td>0.8212</td>
<td>0.8005</td>
<td>0.7915</td>
<td>0.6850</td>
<td>0.5972</td>
<td>0.4316</td>
<td>0.1899</td>
</tr>
<tr>
<td>num_community</td>
<td>6</td>
<td>10</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>14</td>
<td>9</td>
<td>10</td>
<td>45</td>
</tr>
</tbody>
</table>

Graph Partitioning

**Experiment on Dataset2.** Firstly, we display the visual layout of the 146 images which are from 11 pedestrians. These graphs clearly show the cluster property as shown in Fig. 4.
It’s obvious that SC, CAC perform better than SymNMF and all the three methods perform well on Dataset2 as shown in Table 2. In this part, the p-nearest neighbor method is a good method to construct an ideal graph to get an ideal clustering result.

<table>
<thead>
<tr>
<th>Dataset2</th>
<th>AC</th>
<th>MI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sparse_p</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.8411</td>
<td>0.9130</td>
</tr>
<tr>
<td></td>
<td>0.4600</td>
<td>0.9559</td>
</tr>
<tr>
<td>7</td>
<td>0.8767</td>
<td>0.8699</td>
</tr>
<tr>
<td></td>
<td>0.6092</td>
<td>0.9625</td>
</tr>
<tr>
<td>10</td>
<td>0.8219</td>
<td>0.8699</td>
</tr>
<tr>
<td></td>
<td>0.7707</td>
<td>0.8804</td>
</tr>
<tr>
<td>25</td>
<td>0.7055</td>
<td>0.7123</td>
</tr>
<tr>
<td></td>
<td>0.9373</td>
<td>0.8371</td>
</tr>
</tbody>
</table>

Table 2. The graph clustering results of Dataset2.

**Experiment on Dataset3.** Firstly, we display the visual layout of the 837 images which are from 11 pedestrians. These graphs also show obvious cluster property as shown in Fig. 5.

![Visual layout of the similarity graph](image)

Figure 5. The visual layout of the similarity graph of 837 images from 11 pedestrians constructed in different sparse_p value.

The graph clustering results on Dataset3 are showed in Table 3. We can see SC, CAC perform better than SymNMF and all the three methods perform relatively poor. In fact, the difference of the images number of each pedestrian is too large and p-nearest neighbor method is not able to construct an ideal graph to get a good enough clustering result. But the current result is still inspiring and shows strong clustering ability.

<table>
<thead>
<tr>
<th>Dataset3</th>
<th>AC</th>
<th>MI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sparse_p</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.4516</td>
<td>0.7300</td>
</tr>
<tr>
<td></td>
<td>0.1454</td>
<td>0.9251</td>
</tr>
<tr>
<td>10</td>
<td>0.4576</td>
<td>0.6703</td>
</tr>
<tr>
<td></td>
<td>0.2053</td>
<td>0.7304</td>
</tr>
<tr>
<td>15</td>
<td>0.4758</td>
<td>0.5830</td>
</tr>
<tr>
<td></td>
<td>0.2650</td>
<td>0.6953</td>
</tr>
<tr>
<td>20</td>
<td>0.4545</td>
<td>0.5783</td>
</tr>
<tr>
<td></td>
<td>0.3065</td>
<td>0.6277</td>
</tr>
</tbody>
</table>

Table 3. The graph clustering results of Dataset3.
Conclusion and Future Work

We construct a new framework to achieve the task of person re-identification. We calculate the similarity matrix of the pedestrians’ images and construct the similarity graph, then GN algorithm is used to evaluate the number of clusters and partition the graph. The graph partitioning result corresponds to the pedestrians clustering results. Through analyzing each pedestrian clustering result in the experiments, we can see the appearance frequency and which cameras the input person appear in.

There are two aspects that can be explored in the future: First, p-nearest neighbor method is simple and effective to construct a similarity graph when the samples number are almost equal, but it does not perform good any more when the samples number difference are significant. In this condition, we can use other graph construction methods such as LSR [28], LLE [26] to get better pedestrian similarity graph. Second, although GN algorithm usually have a good effect on evaluating the number of clusters, we also notice that its effects are relevant to graph construction. Therefore, it will be meaningful to try some eigenvalue heuristic methods [29].

References


