SHaPE: A Novel Graph Theoretic Algorithm for Making Consensus-based Decisions in Person Re-identification Systems

Arko Barman
University of Houston
4800 Calhoun Road, Houston TX 77004
abarman@uh.edu

Shishir K. Shah
University of Houston
4800 Calhoun Road, Houston TX 77004
sshah@central.uh.edu

Abstract

Person re-identification is a challenge in video-based surveillance where the goal is to identify the same person in different camera views. In recent years, many algorithms have been proposed that approach this problem by designing suitable feature representations for images of persons or by training appropriate distance metrics that learn to distinguish between images of different persons. Aggregating the results from multiple algorithms for person re-identification is a relatively less-explored area of research. In this paper, we formulate an algorithm that maps the ranking process in a person re-identification algorithm to a problem in graph theory. We then extend this formulation to allow for the use of results from multiple algorithms to make a consensus-based decision for the person re-identification problem. The algorithm is unsupervised and takes into account only the matching scores generated by multiple algorithms for creating a consensus of results. Further, we show how the graph theoretic problem can be solved by a two-step process. First, we obtain a rough estimate of the solution using a greedy algorithm. Then, we extend the construction of the proposed graph so that the problem can be efficiently solved by means of Ant Colony Optimization, a heuristic path-searching algorithm for complex graphs. While we present the algorithm in the context of person re-identification, it can potentially be applied to the general problem of ranking items based on a consensus of multiple sets of scores or metric values.

1. Introduction

The application of computer vision in video-based surveillance and forensics has gained much popularity in recent years [31]. Among different challenges within that context, person re-identification has been particularly challenging to solve due to a number of reasons – changes in appearance of a person due to change in viewpoint, pose, illumination, and occlusion. Moreover, the visual appearance of different persons often appear to be similar due to similar clothing or prevalence of common clothing styles.

The problem of person re-identification can be formally stated as – given a gallery of images of a number of persons, a probe image of a person from another viewpoint is to be matched with the corresponding gallery image of the same person. Person re-identification algorithms proposed till date have focused on learning the appearance of different persons [14, 24] and distance metric learning [21, 27]. While designing discriminative features that are able to distinguish between the appearances of different persons [14, 24] is a common feature representation approach, progress has also been made towards extraction of features through the application of deep learning [1, 19].

Performance of person re-identification algorithms has improved steadily over the years. However, the accuracy rates for person re-identification systems are not adequate for implementation in automated video surveillance in the real world. A simple yet effective way to boost the performance of person re-identification systems is to explore the possibility of enhancing the performance of existing algorithms by combining results from multiple algorithms. The utility of this direction of research cannot be overemphasized since the advent of new algorithms will only serve to further enhance the results by feeding results from existing and future algorithms into these aggregating techniques[2].

In this paper, we show how the process of ranking gallery images, in terms of their similarity scores with a probe image, can be mapped to a path searching problem in a graph. This mapping is then extended to include the scenario when there are multiple sets of scores from different algorithms instead of a single set of scores from just one algorithm. This mapping allows us to make decisions on a person re-identification problem based on a consensus of results from multiple algorithms. Finally, we propose a two-step method for solving the formulated graph problem. The overall approach of mapping the ranking process to a graph, and its proposed two-step solution is denoted as SHaPE (Shortest

Hamiltonian Path Estimation). In defining the solution, we make use of a heuristic path-searching algorithm for complex graphs – Ant Colony Optimization (ACO) [11].

The contributions of the paper are summed up as follows:

- We show a novel mapping scheme that maps the process of ranking using scores to a path-searching problem in a graph.
- We extend the mapping scheme to incorporate the use of multiple sets of scores from different algorithms. This results in a problem of consensus-based decision making.
- We propose a two-step method for solving the problem of consensus-based decision making using graph theory. First, a rough estimate is obtained using a greedy algorithm and then the result of the consensus is obtained using Ant Colony Optimization.

We present the results of experiments performed on three of the most commonly used datasets for person re-identification – VIPeR [15], CUHK01 [18] and CUHK03 [19]. The results indicate that the performance of our consensus-based decision-making algorithm results in a significant improvement in performance over individual algorithms. Moreover, our algorithm outperforms other state-of-the-art score fusion techniques used in the context of person re-identification.

2. Related Work

The task of person re-identification begins with feature representation [35, 39]. The goal of the feature representation process is to extract suitable features from the images of different persons in such a manner that the images of the same person have similar features while those of different persons are dissimilar. The process of feature extraction has evolved considerably over the years. The earliest works involve the use of color and texture descriptors include Ensemble of Localized Features (ELF) [15], shape and appearance context modeling [32], and Symmetry Driven Accumulation of Local Features (SDALF) [14]. In more recent years, feature extraction has been done by exploiting properties such as visual saliency [36], custom pictorial structures [6] and the use of regionlets [33]. The use of techniques such as Fisher vector encoding [23], local maximal occurrence representation (LOMO) [21], hierarchical Gaussian descriptors [24], and deep learning [1, 19, 29] have improved the performance of person re-identification markedly in recent years.

Learning a suitable distance metric from the features of annotated images of persons has been an important direction of research. These distance functions are designed and learned in such a way that the images of the same person are "closer" while the images of different persons are "farther apart". Some of the approaches that yield interesting results include the use of Mahalanobis distance [16], RankSVM [28], probabilistic relative distance comparison [38], locally-adaptive decision functions [20], and Cross-view Quadratic Discriminant Analysis (XQDA) [21].

A relatively new direction of research involves the use of existing algorithms to enhance performance [2, 3]. In this context, research has been done to evaluate the improvement in performance by using co-traveler information [34], and exploiting gait features along with existing features [22]. Using the results from many algorithms to make a consensus-based decision has also been explored through the use of rank aggregation [8], fusion of features [13, 17], sum of weighted scores [25], and the use of false alarm rate (FAR) for supervised score fusion [12]. The estimation of posterior probabilities from raw scores for unsupervised fusion [2] and Query-Adaptive late fusion [37] exhibit some of the most promising results in recent years.

3. Mapping the Ranking Process to a Graph

Consider a probe image and a set of n gallery images, which a person re-identification algorithm scores in order to establish a degree of similarity. In our formulation, we assume that the scores are non-negative real numbers. To identify the best match, the scores need to be sorted so that the gallery image with the highest score (similarity) can be established as the best match. This sorting or ranking process can be modeled as follows. Beginning from the gallery image having the lowest similarity score, the ranking process moves towards higher scores until it reaches the highest score. The process of moving towards higher scores, one by one, can be seen as moving to "the nearest score (or gallery image) that has not yet been visited".

The modeling of the ranking process described above can be mapped to a graph, $G = (V, E)$, where V is the set of vertices and E is the set of edges or arcs. In this formulation, we consider the set of vertices to be the set of gallery images, *i.e.*, each vertex corresponds to a gallery image. An edge between any two vertices is undirected and has a weight equal to the absolute difference between the similarity scores of the two gallery images. Consequently, we construct a fully connected graph. In the graph $G = (V, E)$, the set of vertices is $V = \{1, 2, \ldots, n\}$ and the set of edges is $E \subseteq V^2$. An edge between vertices i and j is denoted by (i, j) and the weight associated with (i, j) is denoted as ω_{ij}. Since the graph is fully connected, ω_{ij} exists $\forall\ i, j \in V$ and $i \neq j$. Also, since the graph is undirected and there are no self-loops, $\omega_{ij} = \omega_{ji}$ for $i \neq j$ and $\omega_{ii} = 0\ \forall\ i, j \in V$. We term this graph as a "Score Distance Graph".

The process of ranking can now be translated to finding a suitable path in the Score Distance Graph. The ranking starts with the vertex whose corresponding score is closest

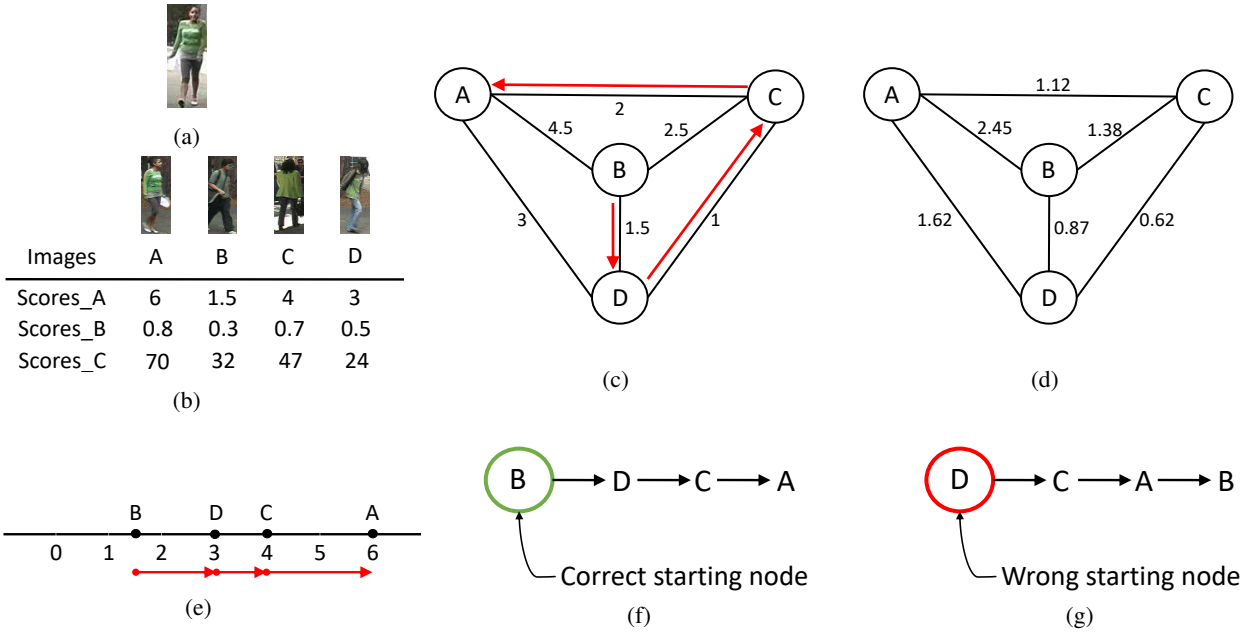


Figure 1: An example of Ranking using a Score Distance Graph for a gallery of 4 images. (a) Probe image, (b) Gallery set along with respective matching scores using 3 different algorithms, (c) Score Distance Graph using scores from just 1 algorithm, Scores_A (Correct ranking path shown in red), (d) Score Distance Graph using scores from all 3 algorithms, (e) Scores plotted on a number line for Scores_A (shortest path connecting all nodes shown in red), (f) Correct ranking, (g) Wrong starting node selection resulting in wrong ranking (worst match chosen as highest ranked).

to 0, and then at each step, moves to the nearest node that has not yet been visited. The process continues until every node has been visited and there are no more nodes left to traverse. The path that has been traversed gives us the ranking of the gallery images based on the similarity scores from lowest to highest. As illustrated through Figure 1, let us assume that a probe image (Figure 1a) is matched against a gallery of 4 images with similarity scores obtained from 3 person re-identification algorithms as shown in Figure 1b. The corresponding Score Distance Graph for only the first algorithm is constructed (Figure 1c) and the starting node is determined to be the node closest to 0 i.e. node B. Figure 1e shows the scores on a number line and B is seen to be the node closest to 0. The shortest path connecting all 4 nodes is shown in red. The correct ranking order is shown in Figure 1f. In fact, this is the path with the smallest sum of edge weights that traverses all the nodes in the Score Distance Graph. Hence, finding this path in the graph would result in ranking the gallery. An incorrectly ranked list formed due to the wrong selection of starting node is shown in Figure 1g.

The path in the graph discussed above traverses all vertices exactly once. Such a path in a graph is known as a Hamiltonian path [9]. A Hamiltonian path in a graph of n vertices can be written as an $(n-1)$-tuple of edge weights,

$$\mathcal{P}_{\mathcal{H}} = \left(\omega_{i_1 i_2}, \omega_{i_2 i_3}, \ldots, \omega_{i_{n-1} i_n}\right) \tag{1}$$

where $i_1, i_2, \ldots, i_n \in E$ are distinct, and $\omega_{i_k i_{k+1}}$ is the weight associated with the edge $(i_k, i_{k+1}) \in V$. In fact, $(i_1, i_2, \ldots, i_n)$ is a permutation of the n vertices in V. The ranking order is given by the nodes on the path written out in reverse order, *i.e.*, $(i_n, i_{n-1}, i_{n-2}, \ldots, i_2, i_1)$.

For the particular case of the process of ranking, the Hamiltonian path found in the graph is actually the shortest Hamiltonian path that starts from the chosen vertex (or, the corresponding gallery image) whose similarity score with respect to the probe is closest to 0. Moreover, it can be seen from the number line in the example shown in Figure 1e that it is one of the two globally shortest possible Hamiltonian paths starting from any node in the graph, the other one having an identical total path length and being just the reverse of this path. The shortest Hamiltonian path among the set of all possible Hamiltonian paths starting from all nodes in a graph is termed as the globally shortest Hamiltonian path in our formulation.

The process of ranking a set of gallery images thus reduces to the problem of finding the globally shortest Hamiltonian path in a graph thus created using the gallery images as the vertices. Either of the two globally shortest Hamiltonian paths is equivalent and give the ranking order for the gallery images. For the sake of simplicity, we will assume that we choose the path that starts from the vertex that has the lowest similarity score and this path gives the ranking in

reverse order.

We now extend the process of ranking using one set of similarity scores to a process of consensus-based decision making using multiple sets of scores. In this case, each vertex still corresponds to a gallery image. However, each gallery image is associated with not just one score but a vector of multiple scores. Each of the similarity scores in the vector is obtained by using a different person re-identification algorithm that computes the similarity between the probe image and that gallery image. If a consensus-based decision is to be made from the scores of K algorithms, then the score vector has a dimension of K. We denote this score vector as an "attribute" of the node.

It is also necessary to suitably define the weights associated with each edge for the Score Distance Graph. An edge weight should be a representative of the "distance" between the score vectors of the two corresponding nodes that it connects. Extending the simple absolute difference method used for scalar scores to Manhattan distance for vectors of scores is not practical since each of the algorithms yields a set of scores with differing ranges and distributions. It is of paramount importance to consider the distribution of scores from each algorithm while calculating the weight of each edge. In order to account for the distributions and variances of the different sets of scores obtained from different algorithms, Mahalanobis distance between the score vectors of respective nodes is used for computing the weight of the edges between any two nodes. For the example shown in Figure 1, the Score Distance Graph constructed using the scores for all 3 algorithms is shown in Figure 1d.

Having constructed the Score Distance Graph, the ranking of gallery images based on the consensus of scores from multiple algorithms can now be found by determining the globally shortest Hamiltonian path in the graph.

4. Shortest Hamiltonian Path Estimation

In this section, we present a two-step solution for estimating the shortest Hamiltonian path in the Score Distance Graph for the purpose of consensus-based decision making. Finding the globally shortest Hamiltonian path in a graph is an optimization problem of finding the path of smallest length that passes through all the nodes in a graph. We propose to solve this optimization problem using Ant Colony Optimization (ACO), a heuristic path-searching algorithm for complex graphs. Some parameters of ACO are dependent on the scale of the problem and need to be initialized suitably. Therefore, the first step is a greedy method which tries to determine the length of the shortest Hamiltonian path in the graph based on the process of ranking for a single set of scores. We then apply Ant Colony Optimization to complete the estimation process.

The simplicity of the ranking process in a single set of scores cannot be simply extended to account for the case of multiple sets of scores. For a single set of scores, the scores can be plotted on a number line (see Figure 1e), and the nodes can be written out in order of their appearance on the number line. However, for a multi-modal scenario, the different distributions of scores from different algorithms are mapped to each dimension, and hence, different non-linearities are introduced across different dimensions. It is, thus, not simply the extension of a number line in many dimensions. For example, the crucial process of selecting the starting node might be flawed, which might result in an incorrect ranking (refer to Figure 1g). Hence, estimating the shortest Hamiltonian path in the Score Distance Graph becomes a non-trivial problem, unlike the simplistic one-dimensional scenario.

4.1. Greedy Nearest Node Search (GNNS)

The starting node for the shortest Hamiltonian path is selected by finding the Mahalanobis distance between every node and a "zero" node with the attribute as a vector of zero values. Having decided the starting node, the subsequent nodes in the path are decided greedily. At every step, the nearest node from the present node is found (based on Mahalanobis distance that is used to compute weights for the edges), and the process continues until all the nodes have been traversed.

As in the notation presented in Equation 1, the starting node is denoted as i_1. The next node i_2 is selected using the following equation,

$$w_{i_1 i_2} = \min_{k \in V \setminus \{i_1\}} w_{i_1 k} \tag{2}$$

where, the set notation $A \setminus B \equiv \{x \in A \mid x \notin B\}$. In a similar manner, the subsequent node in the path, i_3 is selected using the following equation,

$$w_{i_2 i_3} = \min_{k \in V \setminus \{i_1, i_2\}} w_{i_2 k} \tag{3}$$

In general, when the traversal has reached l nodes, the $(l+1)$th node is selected from the graph using the following equation,

$$w_{i_l i_{l+1}} = \min_{k \in V \setminus \{i_1, i_2, \ldots, i_l\}} w_{i_l k} \tag{4}$$

After traversing all nodes, the path $(i_1, i_2, \ldots, i_n)$ gives a rough estimate of the ranking in reverse order. The ranking order generated by the greedy algorithm, is thus, given by the n-tuple,

$$\mathcal{R} = (i_n, i_{n-1}, i_{n-2}, \ldots, i_2, i_1) \tag{5}$$

The length of the path obtained from this greedy nearest node search is denoted as L_{nn} and is used to initialize certain parameters in the ACO step for solving this problem.

4.2. Letting Ants Find the Way

As the second step for our two-step solution to the consensus-based ranking problem, we use Ant Colony Optimization [10, 11]. Ant Colony Optimization (ACO) is a metaheuristic that has been widely applied for solving discrete optimization problems, especially for finding paths in a complex graph based on a specified set of criteria.

In order to apply ACO for solving the path search problem in the graph in a heuristic fashion, the Score Distance Graph that we proposed in Section 4 needs to be slightly modified. We convert the problem of finding the shortest Hamiltonian path in the n-node Score Distance Graph to an equivalent Traveling Salesman Problem (TSP) in an $(n+1)$-node graph [5]. A "dummy" node is added to the n-node graph G to obtain an $(n+1)$-node graph, G'. It is imperative to note here that this "dummy" node is different from the "zero" node used previously. The zero node is not used to extend the graph, but just to estimate the starting node for greedy nearest node search. While the attributes of the zero node are all zeros, the attributes of the dummy node are "don't care", since we do not really need to consider the attributes of the dummy node for computations. For this step of the solution, we require the use of the zero node only once more at the end. The "dummy" node is connected to every node in G, so that the properties of the graph G described in Section 2 also hold for the graph G'. The weight of every edge connecting the dummy node is 0. This ensures that a circuit (a circuit in a graph is a path that ends at the same node where it begins) can be found in G' without affecting the cost of traversing through the dummy node: the ends of the shortest Hamiltonian path can be connected to each other through the dummy node without adding to the length of the path, thereby getting a solution for the TSP. Solving the TSP for G' is now equivalent to solving for the shortest Hamiltonian path in G. The solution of the TSP yields the shortest Hamiltonian circuit (a circuit that passes through all nodes exactly once). The starting and ending nodes in the path can be identified by checking the distance between the two nodes that are connected to the dummy node with the zero node. The one with the lower Mahalanobis distance gives the starting node, and the ranking order of the gallery images can be written as the reverse of the path that was thus found.

4.2.1 How do ants find the shortest path?

It has been observed that ants efficiently find the shortest path between their nest (source) and a food source (destination) [11]. While traversing paths between the source and destination, ants deposit chemical factors known as pheromones. Since ants that follow shorter paths return to the source (from a round-trip to the destination) faster than the ones that follow longer paths, the pheromone deposited on shorter paths increase much faster compared to longer paths. Ants that have to make decisions between competing paths at a future time, choose the path with more pheromone with a greater probability than the paths with lesser pheromones. As this process continues, most ants in the colony converge to the best (shortest) path between the source and the destination. The ants, thus, work in "parallel" and "cooperate through pheromone-mediated indirect and global communication" [10].

Every edge e_i in the graph is assigned a parameter τ_i known as artificial pheromone trail or just pheromone trail. In the beginning, all edges have the same pheromone trail, c. Let the cost of a path or sequence s be denoted as $f(s)$ and the search space as $\mathcal{S}$. Each ant builds a solution, starting from an empty sequence, $s = \langle\rangle$. At each solution construction step, an ant extends the current sequence s by moving to an unvisited node, *i.e.* it selects an edge from the unvisited neighborhood of the current node, denoted by $\mathcal{N}(s) \subseteq E \setminus s$. An ant selects an unvisited node from $\mathcal{N}(s)$ using "transition probabilities" which are, in turn, dependent on the values of the pheromone trail of the edges according to the following equation [4],

$$\mathbf{p}(e_i \mid s) = \frac{[\tau_i]^\alpha \cdot [\eta(e_i)]^\beta}{\sum_{e_j \in \mathcal{N}(s)} [\tau_j]^\alpha \cdot [\eta(e_j)]^\beta} \tag{6}$$

where τ_i is the pheromone trail on the edge e_i, and η is a weighting function that assigns each edge with a "heuristic value" [11], usually set to be the inverse of the edge weight (in case of edge weight being zero, the edge weight is replaced by a very small value during calculation). The parameters α and β are set to 1 and 2 respectively, based on experiments involving real ants [11]. Contrary to what is observed in real ants, pheromone is not deposited by ants while moving in the forward direction (for problems other than TSP, deposition of pheromone in the forward mode often results in creation of self-reinforcing loops).

On reaching the destination, each ant has created a complete sequence from source to destination and it now switches to backward mode. In this mode, they traverse the path in the reverse direction, depositing pheromone on the edges that they traversed in their forward path. Pheromone update is performed in such a way that shorter paths are deposited with more pheromone than longer paths. Moreover, pheromone evaporation is done in order to discourage early sub-optimal convergence. In the backward mode, pheromone update is performed on edges on the forward path according to the following equation [4],

$$\tau_i \leftarrow (1-\rho) \cdot \tau_i + \rho \cdot \sum_{\{s \in \mathcal{S}_{upd} | e_i \in s\}} w_s \cdot F(s) \tag{7}$$

for $i = 1, 2, \ldots, n$. Here, $\rho \in (0, 1]$ is the evaporation rate of pheromone, $\mathcal{S}_{upd}$ is the set of solutions that

are used for the update, and $F : \mathcal{S} \mapsto \mathbb{R}^+$ is a quality function, that decides the quality of the path the ant had taken in the forward direction, and is chosen such that $f(s) < f(s') \implies F(s) \geq F(s') \; \forall \; s \neq s'$ and $s, s' \in \mathcal{S}$. This ensures that the quality of a solution with shorter path length is at least as much as that of a longer path, and possibly larger. An additional weight, $w_s \in \mathbb{R}^+$ is used for scaling the quality function.

The set of solutions used for the update, $\mathcal{S}_{upd}$, consists of either the solutions generated in the respective iteration ($\mathcal{S}_{iter}$) or the best solution found since the first iteration (denoted by s_{bs}) or both. We use an update rule known as *BS-update* (Best-So-far update) [4], where only s_{bs} is used for updating the pheromone values and $\mathcal{S}_{iter}$ is ignored. Hence, $\mathcal{S}_{upd} = \{s_{bs}\}$, and the weight, w_s, is set to 1.

4.2.2 Ant Colony System

Considering the performance of different variants of ACO, an improved algorithm known as Ant Colony System (ACS) [10], with better performance and faster convergence than the traditional ACO algorithm, is used in our experiments. In this algorithm, an ant chooses the next edge that maximizes $[\tau_i]^\alpha \cdot [\eta\,(e_i)]^\beta$ with a probability q_0 (exploitation), or uses Equation 6 with a probability $(1 - q_0)$ (exploration). The parameter q_0 decides the relative balance between exploitation and exploration. This solution construction is termed as *pseudo-random proportional*. The pheromone update rule *BS-update* is followed with a small modification. Only if an ant constructs a best-so-far path, it is allowed to update pheromone while other ants do not update pheromone. The quality function for pheromone update is the inverse of the path length for the best-so-far path. Additionally, at the end of each solution construction step, the edge that is added to the sequence under construction undergoes a pheromone update,

$$\tau_i = (1 - \xi) \cdot \tau_i + \xi \cdot \tau_0 \tag{8}$$

where $\xi \in (0, 1)$ is a parameter that determines how much the pheromone trail is decreased, τ_0 is a small positive constant that satisfies the condition $c \leq \tau_0 \leq F_{min}$, $F_{min} \leftarrow \min\{F(s) \mid s \in \mathcal{S}\}$, and c is the initial value of the pheromone trail. This pheromone update leads to a decrease in pheromone values of the traversed edges, thereby making these edges less desirable to ants that arrive at that particular node at a later time step in the respective iteration. As a result, this mechanism encourages the exploration of paths that have been less traversed within an iteration, since traversed edges have reduced pheromone trails.

4.2.3 Parameters

The parameters for ACS have the following values in our experiments: $\alpha = 1$, $\beta = 2$, $q_0 = 0.9$, $\rho = \xi = 0.1$, and $\tau_0 = ((n + 1) \cdot L_{nn})^{-1}$, where $(n + 1)$ is the number of nodes and L_{nn} is the length of the shortest Hamiltonian path as estimated by the greedy nearest node search. The parameter, τ_0, is thus dependent on the scale of the problem, unlike the other parameters [10]. The number of ants used is 10. Initially, each ant is placed randomly on a node, with no node having more than one ant.

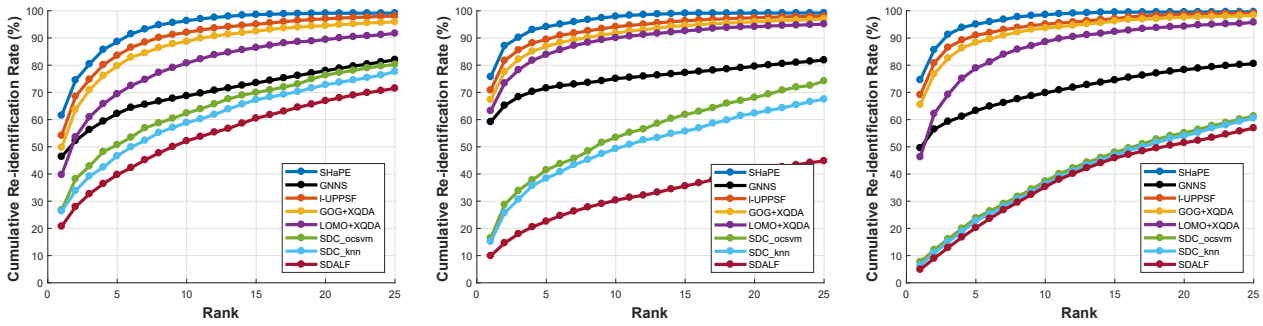
The entire two-step solution for Shortest Hamiltonian Path Estimation (SHaPE) is shown in Algorithm 1.

Algorithm 1 SHaPE Algorithm

Find Mahalanobis distance between each pair of score vectors (i.e. gallery images)
Construct Score Distance Graph (vertex $\equiv$ gallery image, edge $\equiv$ distance between score vectors)
Create a "zero" node with all attributes 0
procedure GREEDY NEAREST NODE SEARCH (GNNS)
    Select estimated starting node by finding the node closest to "zero" node
    while (unvisited nodes are present) **do**
        Choose next node on path using Equation 4
    end while
    Compute the length of the greedy path, L_{nn}
end procedure
procedure ANT COLONY SYSTEM
    Add a "dummy node" with edges to all nodes having 0 weight
    Initialize parameters, $\alpha = 1$, $\beta = 2$, $q_0 = 0.9$, $\rho = \xi = 0.1$, and $\tau_0 = ((n + 1) \cdot L_{nn})^{-1}$
    while (Convergence is not reached) **do**
        Allow ants to construct solutions
        Update pheromone trails
    end while
end procedure
Consider the 2 nodes in the TSP solution connected to "dummy node": the node nearer to "zero" node is the starting point for the reverse ranked list

5. Experiments

In order to evaluate the performance of SHaPE algorithm for consensus-based decision-making, experiments were performed on three benchmark datasets: VIPeR [15], CUHK01 [18] and CUHK03 [19]. For the purpose of comparison, we have used Cumulative Matching Characteristics (CMC), which is a plot of the cumulative re-identification rate for increasing ranks. In our experiments, we generate a consensus of results obtained from state-of-the-art person re-identification algorithms – SDALF [14], SDC_knn [36], SDC_ocsvm [36], LOMO+XQDA [21], and $\text{GOG}_{\text{Fusion}}$+XQDA [24]. Our experiments show that the re-



(a) VIPeR (b) CUHK01 (c) CUHK03

Figure 2: Performance of SHaPE in making consensus-based decisions for (a) VIPeR, (b) CUHK01, and (c) CUHK03. The performance of l-UPPSF [2] for score fusion is also shown here. Intermediate results for GNNS are shown for comparison with SHaPE.

Method	Rank-1	Rank-2	Rank-3	Rank-5	Rank-10	Rank-15	Rank-20
CH [37]	22.86%	30.73%	35.71%	42.72%	53.91%	61.38%	65.98%
CN [30]	21.74%	28.67%	33.96%	41.41%	50.24%	55.87%	60.85%
LBP [26]	5.73%	9.46%	12.09%	15.52%	23.59%	29.26%	34.19%
HOG [7]	5.40%	7.91%	9.73%	12.75%	19.42%	24.38%	28.20%
SDC_ocsvm [36]	23.78%	32.82%	38.23%	45.70%	57.48%	65.46%	71.08%
QAF [37]	30.17%	38.61%	43.82%	51.60%	62.44%	69.06%	73.81%
l-UPPSF [2]	30.35%	39.40%	45.13%	53.35%	64.08%	71.23%	76.23%
SHaPE	**34.26%**	**42.97%**	**48.29%**	**57.34%**	**67.86%**	**75.16%**	**80.78%**

Table 1: Comparison of cumulative person re-identification rates for VIPeR dataset at different ranks using Color Histograms (CH), Color Names (CN), Local Binary Patterns (LBP), Histogram of Oriented Gradients (HOG), SDC_ocsvm, and their consensus using Query-adaptive Fusion, l-UPPSF and SHaPE. The figures in bold indicate the best performance for the respective rank.

Dataset (Gallery size)	VIPeR ($N = 316$)	CUHK01 ($N = 486$)	CUHK03 ($N = 100$)
Execution time (in seconds)	63.17	95.29	13.76

Table 2: Computation times for Matlab implementation of SHaPE on a 1.7GHz Intel Core i3 processor with 8GB RAM for different datasets and gallery sizes.

sults of person re-identification generated by the consensus significantly surpass the results of each of the algorithms used individually.

Further, we compare the performance of SHaPE with the results of two state-of-the-art score fusion algorithms – l-UPPSF [2] and Query-Adaptive Fusion (QAF) [37]. It can be seen from the results that SHaPE achieves a better performance at combining results from multiple algorithms compared to these algorithms.

The VIPeR dataset [15] is one of the most commonly used datasets for analyzing the performance of person re-identification algorithms. The dataset contains 2 images each of 632 persons, each from a different camera view. Change in illumination between two images of the same person as well as the large change in viewing angle poses a considerable challenge for re-identification. The dataset is split into half with images of 316 persons being used for testing. Experiments were conducted using 10 such splits as reported in [14]. A consensus of results from SDALF [14], SDC_knn [36], SDC_ocsvm [36], LOMO+XQDA [21], and $\text{GOG}_{\text{Fusion}}$+XQDA [24] was done, and the results were compared with l-UPPSF [2]. The CMC curves obtained from the experiments is shown in Figure 2a. The CMC curves show that using SHaPE for making consensus-based decisions results in a much higher accuracy than using state-of-the-art algorithms individually. Moreover, the performance of SHaPE also supersedes the performance of l-UPPSF, a state-of-the-art score fusion method.

The CUHK01 dataset [18] contains 4 images each of 971 persons, with 2 images of every person taken from one camera while the other 2 images are taken from another camera view. For CUHK01 dataset, we tested the performance of the SHaPE algorithm in combining the results of SDALF [14], SDC_knn [36], SDC_ocsvm [36], LOMO+XQDA [21], and $\text{GOG}_{\text{Fusion}}$+XQDA [24], using a multi-shot setting with $M = 2$, as reported in [21] and [24]. In other words, both the images of 486 persons (the rest of the images are used for training by the supervised algorithms – SDC_knn, SDC_ocsvm, LOMO+XQDA,

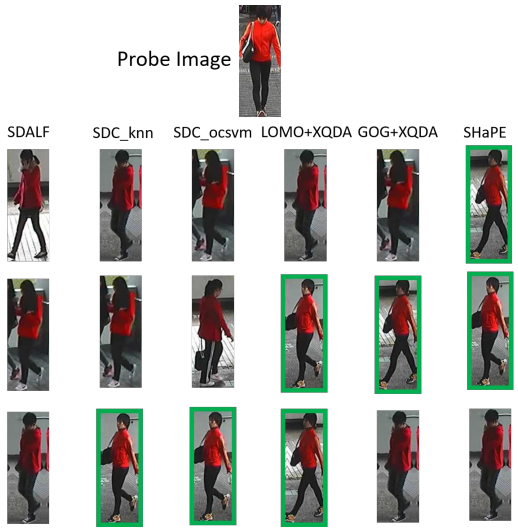


Figure 3: A person re-identification example showing top 3 matches for a sample probe image from CUHK01 dataset. Correct matches are shown in a green bounding box.

and $\text{GOG}_{\text{Fusion}}$+XQDA) from the second camera view were used for constructing the gallery set. Experiments were performed on 10 such splits in the dataset, in the same way as the experiments for the VIPeR dataset. The CMC curves for all the individual algorithms and their consensus using SHaPE is shown in Figure 2b. Further, the performance of l-UPPSF [2] in combining the results of the individual algorithms is also shown. The CMC curves show a significant improvement in performance not only over the individual algorithms, but also over l-UPPSF.

The CUHK03 dataset [19] comprises two sets of annotations for images of persons. One set has been obtained by manually drawing bounding boxes around the images of persons, while the other set has been automatically detected using a pedestrian detector algorithm. Since the automatically detected bounding boxes provide a more realistic approach to the problem of person re-identification in real life, our experiments have been conducted using this set of images. Using this set of images inherently introduces challenges that are not encountered in most other datasets – presence of background clutter, misalignments, missing body parts, and the presence of other persons in a bounding box. The images of 1360 persons were captured from two camera views for a total of 13, 164 images. Each person has an average of 4.8 images in the dataset. Using the same experimental setup in [24], the gallery set consists of 100 persons, and the experiments are conducted for 20 such gallery sets. The performance of SHaPE in making consensus-based decisions using SDALF [14], SDC_knn [36], SDC_ocsvm [36], LOMO+XQDA [21], and $\text{GOG}_{\text{Fusion}}$+XQDA [24] is shown in Figure 2c. The performance of l-UPPSF in combining scores from these algorithms is also shown in the figure. A marked improvement in performance can be observed over the individual re-identification algorithms and their fusion using l-UPPSF.

That the Greedy Nearest Node Search (GNNS) is inadequate in estimating the shortest Hamiltonian path in the graph can be seen in Figure 2 by comparing the CMC curves for GNNS and SHaPE. It can be clearly seen that due to the non-linearities introduced in different dimensions and a possibly flawed selection of the initial node, the results from GNNS are far inferior compared to the final results of the entire SHaPE algorithm (including the ACS solution).

The computation times for a Matlab implementation of SHaPE run on a computer with 1.7GHz Intel Core i3 processor with 8GB RAM are shown in Table 2 for all 3 datasets used in our experiments. An example probe image with the top 3 matches for re-identification results using 5 individual algorithms and their consensus using SHaPE for CUHK01 dataset is shown in Figure 3.

Lastly, we compare the performance of SHaPE with Query-Adaptive Fusion (QAF) [37] using the same experiments in [37]. Query-Adaptive Fusion combines the results of person re-identification for VIPeR dataset using the following features individually (with a Bag-of-Words model) – 1. 20-dimensional Color Histograms (CH) using Hue and Saturation [37]; 2. 11-dimensional Color Names (CN) [30]; 3. Local Binary Patterns (LBP) [26]; 4. Histograms of Oriented Gradients (HOG) [7]; and 5. SDC_ocsvm [36]. The performance of SHaPE in obtaining consensus-based decisions using these features is shown in Table 1. Additionally, the performance of l-UPPSF [2] is also shown in the table. It can be seen from Table 1 that the performance of SHaPE is superior to both QAF and l-UPPSF.

6. Conclusion

In this paper, we have shown a novel mapping of the process of ranking to a graph based on a set of scores. We then show how this process in a graph can be extended for multiple sets of scores in order to make a consensus-based decision. The problem of making consensus-based decisions using scores from different algorithms has been converted to a problem in graph theory. We have proposed a two-step solution of this problem by first employing a greedy nearest node search to obtain an approximate solution, which is then used to initialize certain parameters in the next step of the solution that applies Ant Colony Optimization, a heuristic algorithm for finding paths in graphs based on a specified set of criteria. Since the process of mapping and solving the problem of consensus-based decision-making using a graph can be widely applied to other problems as well, our future work will involve the application of the proposed SHaPE algorithm to other problem domains.

References

[1] E. Ahmed, M. Jones, and T. K. Marks. An improved deep learning architecture for person re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[2] A. Barman and S. K. Shah. Improving person re-identification systems: A novel score fusion framework for rank-n recognition. In *Proceedings of the Tenth Indian Conference on Computer Vision, Graphics and Image Processing*, ICVGIP '16, pages 4:1–4:8, New York, NY, USA, 2016. ACM.

[3] A. Barman and S. K. Shah. Distance aggregation based score fusion for improving person re-identification. In *2017 IEEE International Symposium on Technologies for Homeland Security (HST)*, pages 1–8, April 2017.

[4] C. Blum. Ant colony optimization: Introduction and recent trends. *Physics of Life Reviews*, 2(4):353 – 373, 2005.

[5] D.-S. Chen, R. G. Batson, and Y. Dang. *Applied Integer Programming: Modeling and Solution*. John Wiley & Sons, Hoboken, NJ, 2011.

[6] D. S. Cheng, M. Cristani, M. Stoppa, L. Bazzani, and V. Murino. Custom pictorial structures for re-identification. In *British Machine Vision Conference (BMVC)*, 2011.

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893 vol. 1, June 2005.

[8] R. F. de Carvalho Prates and W. R. Schwartz. CBRA: color-based ranking aggregation for person re-identification. In *2015 IEEE International Conference on Image Processing, ICIP 2015, Quebec City, QC, Canada, September 27-30, 2015*, pages 1975–1979, 2015.

[9] R. Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

[10] M. Dorigo and L. M. Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, Apr 1997.

[11] M. Dorigo and T. Stützle. *Ant Colony Optimization*. Bradford Company, Scituate, MA, USA, 2004.

[12] M. Eisenbach, A. Kolarow, K. Schenk, K. Debes, and H. M. Gross. View invariant appearance-based person reidentification using fast online feature selection and score level fusion. In *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, pages 184–190, Sept 2012.

[13] M. Eisenbach, A. Kolarow, A. Vorndran, J. Niebling, and H. Gross. Evaluation of multi feature fusion at score-level for appearance-based person re-identification. In *2015 International Joint Conference on Neural Networks, IJCNN 2015, Killarney, Ireland, July 12-17, 2015*, pages 1–8, 2015.

[14] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani. Person re-identification by symmetry-driven accumulation of local features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2360 –2367, June 2010.

[15] D. Gray and H. Tao. *Viewpoint Invariant Pedestrian Recognition with an Ensemble of Localized Features*, pages 262–275. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[16] M. Hirzer, P. M. Roth, M. Köstinger, and H. Bischof. *Relaxed Pairwise Learned Metric for Person Re-identification*, pages 780–793. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[17] Y. Hu, S. Liao, Z. Lei, D. Yi, and S. Z. Li. Exploring structural information and fusing multiple features for person re-identification. In *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 794–799, June 2013.

[18] W. Li, R. Zhao, and X. Wang. Human reidentification with transferred metric learning. In *ACCV*, 2012.

[19] W. Li, R. Zhao, T. Xiao, and X. Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 152–159, 2014.

[20] Z. Li, S. Chang, F. Liang, T. S. Huang, L. Cao, and J. R. Smith. Learning locally-adaptive decision functions for person verification. In *Proceedings of CVPR*, 2013.

[21] S. Liao, Y. Hu, X. Zhu, and S. Z. Li. Person re-identification by local maximal occurrence representation and metric learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[22] Z. Liu, Z. Zhang, Q. Wu, and Y. Wang. Enhancing person re-identification by integrating gait biometric. *Neurocomputing*, 168:1144 – 1156, 2015.

[23] B. Ma, Y. Su, and F. Jurie. *Local Descriptors Encoded by Fisher Vectors for Person Re-identification*, pages 413–422. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[24] T. Matsukawa, T. Okabe, E. Suzuki, and Y. Sato. Hierarchical gaussian descriptor for person re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[25] L. Nanni, M. Munaro, S. Ghidoni, E. Menegatti, and S. Brahnam. Ensemble of different approaches for a reliable person re-identification system. *Applied Computing and Informatics*, pages –, 2015.

[26] T. Ojala, M. Pietikinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51 – 59, 1996.

[27] S. Paisitkriangkrai, C. Shen, and A. van den Hengel. Learning to rank in person re-identification with metric ensembles. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 1846–1855, 2015.

[28] B. Prosser, W.-S. Zheng, S. Gong, and T. Xiang. Person re-identification by support vector ranking. In *Proceedings of the British Machine Vision Conference*, pages 21.1–21.11. BMVA Press, 2010. doi:10.5244/C.24.21.

[29] H. Shi, Y. Yang, X. Zhu, S. Liao, Z. Lei, W. Zheng, and S. Z. Li. *Embedding Deep Metric for Person Re-identification: A Study Against Large Variations*, pages 732–748. Springer International Publishing, Cham, 2016.

[30] J. van de Weijer, C. Schmid, and J. Verbeek. Learning color names from real-world images. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, June 2007.

[31] R. Vezzani, D. Baltieri, and R. Cucchiara. People reidentification in surveillance and forensics: A survey. *ACM Comput. Surv.*, 46(2):29:1–29:37, Dec. 2013.

[32] X. Wang, G. Doretto, T. Sebastian, J. Rittscher, and P. Tu. Shape and appearance context modeling. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, Oct 2007.

[33] X. Wang, M. Yang, S. Zhu, and Y. Lin. Regionlets for generic object detection. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[34] L. Wei and S. K. Shah. Subject centric group feature for person re-identification. In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 28–35, June 2015.

[35] Y. Xu, L. Lin, W.-S. Zheng, and X. Liu. Human re-identification by matching compositional template with cluster sampling. In *International Conference on Computer Vision (ICCV)*, Sydney, Australia, December 2013.

[36] R. Zhao, W. Ouyang, and X. Wang. Unsupervised salience learning for person re-identification. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3586–3593, June 2013.

[37] L. Zheng, S. Wang, L. Tian, F. He, Z. Liu, and Q. Tian. Query-adaptive late fusion for image search and person re-identification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[38] W.-S. Zheng, S. Gong, and T. Xiang. Person re-identification by probabilistic relative distance comparison. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 649–656, Washington, DC, USA, 2011. IEEE Computer Society.

[39] W. S. Zheng, X. Li, T. Xiang, S. Liao, J. Lai, and S. Gong. Partial person re-identification. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4678–4686, Dec 2015.