Pattern Recognition 47 (2014) 1126–1137

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Object tracking across non-overlapping views by learning inter-camera transfer models ☆

Xiaotang Chen *, Kaiqi Huang, Tieniu Tan

Center for Research on Intelligent Perception and Computing, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, No.95 ZhongGuanCun East Street, HaiDian District, P.O. Box 2778, Beijing, PR China

ARTICLE INFO

Available online 24 June 2013

Keywords:
Object tracking
Transfer models
Color transfer
Camera network
Non-overlapping views

ABSTRACT

In this paper, we introduce a novel algorithm to solve the problem of object tracking across multiple non-overlapping cameras by learning inter-camera transfer models. The transfer models are divided into two parts according to different kinds of cues, i.e. spatio-temporal cues and appearance cues. To learn spatio-temporal transfer models across cameras, an unsupervised topology recovering approach based on N-neighbor accumulated cross-correlations is proposed, which estimates the topology of a non-overlapping multi-camera network. Different from previous methods, the proposed topology recovering method can deal with large amounts of data without considering the size of time window. To learn inter-camera appearance transfer models, a color transfer method is used to model the changes of color characteristics across cameras, which has an advantage of low requirements to training samples, making update efficient when illumination conditions change. The experiments are performed on different datasets. Experimental results demonstrate the effectiveness of the proposed algorithm.

1. Introduction

As the number of cameras used in the wide area video surveillance increases, multi-camera object tracking plays a more important role in understanding and analyzing the scenes. It is a challenging problem. Especially when there are non-overlapping views among cameras, the observations of the same object under different cameras are often widely separated in time and space. Based on this fact, the problem of object tracking across non-overlapping cameras is quite different from single camera object tracking or overlapping multi-camera object tracking.

Over the last few years, many approaches have been proposed to solve this problem. Two kinds of cues are usually employed: spatio-temporal cues across cameras and appearance cues of objects.

1.1. Spatio-temporal cues across cameras

To model the spatio-temporal relationships across cameras, various strategies are proposed to recover the topology graph of the non-overlapping multi-camera network. Fig. 1 shows a topology graph of a non-overlapping multi-camera network. The topology graph usually has three main factors: firstly, the nodes, from which objects enter or exit; secondly, the links between nodes, indicating the connectivity of each two nodes and corresponding to the real paths in the environment which can be followed by objects; thirdly, the transition time distribution for each link across cameras, demonstrating the probability of transition time of an object moving from one node to another. If an object leaves the FoV of a camera at a moment, then we can predict the object's re-appearance after some time under certain cameras using the knowledge of topology.

Generally, the nodes are defined as entry/exit zones in the FoVs of cameras, which can be learned by clustering the starting or ending points of trajectories observed by single camera tracking [2–4], or defined as single cameras [5]. To estimate the existence of link between two nodes and the transition time distribution for each link, the methods can be put into two categories. The first one is based on solving the correspondence problem [3,6,7] or object tracking [8]. Javed et al. [6] use Parzen windows to estimate the inter-camera space–time probabilities from training data, assuming the correspondences are known. These methods usually have good estimations of the transition time distributions, however, solving the problem of correspondences or object tracking itself is complicated and challenging.

The second one does not require establishing correspondences between observations or object tracking [2,4,9,10]. Makris et al. [9] calculate a cross-correlation function of two signals which represent the arrival event sequence observed at one node and the departure event sequence observed at the other node in a time

☆A preliminary work of this paper is published in [1].

* Corresponding author. Tel.: +86 10 62653768; fax: +86 10 62551993.
E-mail addresses: xtchen@nlpr.ia.ac.cn, chenxiaotang@163.com (X. Chen), kqhuang@nlpr.ia.ac.cn (K. Huang), tnt@nlpr.ia.ac.cn (T. Tan).

0031-3203/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2013.06.011

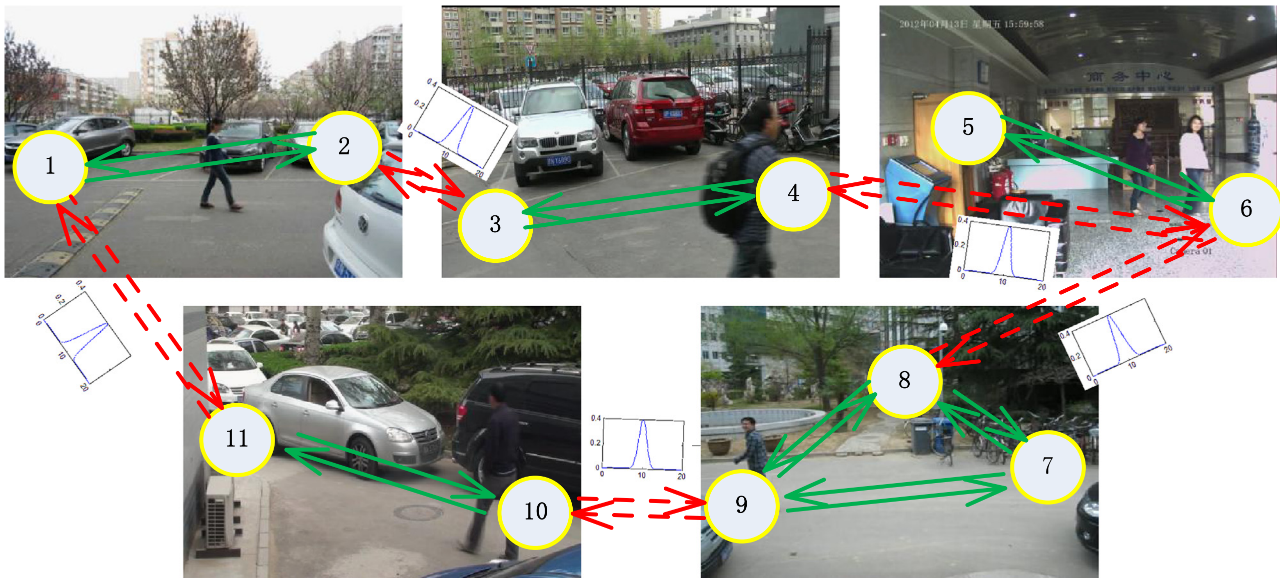

Fig. 1. The topology graph of a non-overlapping multi-camera network. Nodes are entry/exit zones labeled by different numbers. The green solid arrows denote visible paths within the field of view (FoV) of each camera, which can be detected by single camera tracking. The red dotted arrows represent valid links between nodes across cameras, which depend on methods of recovering the topology to estimate the existence and corresponding transition time distributions. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

window. Ideally if a link exists, then the cross-correlation has a clear peak around the most popular transition time. However, in most cases, the peak is not so clear due to the large variance of transition time of true correspondences and a large number of false correspondences which result from a large traffic flow or a long time window. To make the peak sharp, methods [4,10] add similarity in appearance to weight the cross-correlation model. These methods are usually easy to be implemented. However, few of them consider the estimation of transition time distributions. To estimate the transition time distribution, Zou et al. [10] fit K Gaussian functions to a normalized cross-correlation using the EM algorithm, which is not proper for considering both true and false correspondences.

Based on cross-correlation functions, we present a topology recovering method by decreasing the large variance of transition time of true correspondences, which can compensate for the influence caused by large-scale false correspondences to a certain degree. Thus, the proposed topology recovering method can deal with large amounts of data or a long time window. It estimates the transition time distribution for each valid link based on an iteration, which is different from other cross-correlation based methods. In addition, the proposed topology recovering method avoids solving the problem of establishing correspondences between non-overlapping views, making it easy to be implemented.

1.2. Appearance cues of objects

For the appearance cues, methods generally use one or multiple kinds of features to represent the appearance of an object. However, the appearance varies a lot across cameras, which is influenced by many factors, such as the illumination, camera properties, viewpoints, poses and nonuniform clothing, as shown in Fig. 2. Various appearance descriptors are proposed to be robust to the challenges mentioned above. A major color spectrum histogram (MCSH) [11] is introduced to represent a moving object by using a normalized geometric distance between two points in the RGB space. D'Angelo et al. present a probabilistic color histogram (PCH) to describe the color appearance of the object [12], which is built by using a fuzzy K-Nearest Neighbors classifier. Yu et al. [13] model the appearance based on spatial/color statistical features. To incorporate structural information and achieve invariance to motion and pose, they use an additional feature of path-length besides color features. Wang et al. [14] introduce the concept of shape and appearance context by modeling the spatial distribution of the appearance relative to each of the object parts. Instead of directly exploiting robust features in object representation, several approaches [15–17] use machine learning tools (i.e. AdaBoost algorithm) to learn the similarity or distance between any two objects based on color and texture histogram features.

To the best of our knowledge, color-based features are the most widely used features in solving the problem of tracking pedestrians across non-overlapping cameras. Colors are easily influenced by illumination changes. Thus, alleviating the influence of illumination variance across cameras becomes necessary and important. Methods to this problem can be divided into two groups. The first group is color transfer across cameras. One of the typical methods is learning brightness transfer functions (BTFs) [18], which handles the change in observed colors of an object as it moves from one camera to another. Javed et al. [18] show that all brightness transfer functions from a given camera to another camera lie in a low dimensional subspace and demonstrate that this subspace can be used to compute appearance similarity. They learn BTFs for each pair of cameras from the training data by using probabilistic principal component analysis. However, their method relies on training subjects with a good range of brightness values to give an accurate mean BTF (MBTF), which implicitly assumes both extensive color variations on object clothing and very large number of objects being sampled [19]. To extend the work in [18], Prosser et al. [19] compute a cumulative BTF (CBTF) by accumulating the brightness values of the whole training set before the BTF computation instead of computing a BTF for each training pair, thus, brightness values that are not common in the training set are still preserved. Mazzeo et al. [20] compare the performance of two different brightness transfer functions, i.e. the MBTF and the CBTF, which demonstrates quite similar behaviors of the two methods when the simple association problem has to be solved. Jeong et al. [21] learn color transfer functions by operating