Inter-Camera Color Calibration for Object Re-identification and Tracking

Atif Ilyas, Mihaela Scuturici, Serge Miguet
Université de Lyon, CNRS
Université Lumière Lyon 2, LIRIS, UMR5205
Lyon, France

Abstract—Object re-identification and tracking in non-overlapping cameras is a challenging problem due to the variation of the object’s appearance, linked to the different view angle, distance and color variation in different cameras. We present a computationally efficient real time human tracking algorithm, which can track objects inside the field of view (FOV) of a camera, re-identify objects that exit and then return in a same or in a different camera FOV. Object appearance in several cameras may be very different due to illumination conditions, camera gain, focus, focal length etc. Therefore inter-camera color calibration is important for object re-identification before applying object recognition features. We compare different existing color calibration methods and evaluate their color brightness transfer function (BTF) using Receiver Operating Characteristic (ROC) curve. We propose some modifications in cumulative brightness transfer function (CBTF) which significantly improve the objects re-identification in non-overlapping multi camera environment.

Keywords—video surveillance; brightness transfer function; color calibration; object tracking; object identification;

I. INTRODUCTION

Active surveillance systems are used to detect unauthorized or suspicious behaviors and then register the activities of objects and their trajectories. In realistic scenario, large area surveillance is done by using non overlapping cameras. In this paper, we investigate object identification and tracking in multiple non overlapping cameras separated by blind regions. We have previously proposed [1] an appearance based object tracking algorithm having the ability to re-identify the moving objects. Our tracking process is applied independently to each single camera, but the objects features are stored in a common object database which allows object recognition when the object exits from the field of view of one camera and then enters in another cameras. The system performance is not affected by the distance between moving objects and cameras, (the object may be close to the camera - big object - or far away - small object) and it can also be applied to indoor as well as outdoor scenes. Objects colors appearance are very different in different cameras due to cameras model, gain and the region where they are installed. Due to the above discussed reasons, cameras color calibration is essential. Existing algorithms are unable to perform reliable real time object recognition when objects are deformable and exit and re-enter in the field of view of the cameras. Some of extensive survey work on the deformable object detection, recognition and tracking are presented in [2]–[4]. In these surveys, author discuss various algorithms from each class and explain their advantages and limitations. Some related work on 1-D color appearance models are presented in [5]–[7]. K. Sato et al. [5] discuss many aspects of object tracking and interaction. They use 1-D appearance model with geometrical features like horizontal projection and horizontal size for object matching. Therefore, it is less reliable for human tracking as the apparent size can be very different in different frames or in different camera’s field of view. A. Elgammal et al. [6] divide objects height into three parts: head, torso and bottom. Division of object height into only three part is not sufficient to capture the object’s vertical color variation. A. Mittal et al. [7] use overlapping multi-camera system for human tracking. They divide object height into h slices and model each slice with a Gaussian kernel as explained in [6]. Using of N Gaussians for each h slices increases its computational cost.

Inter-camera color calibration plays a vital role in appearance based object identification and tracking. Porikli et al [8] compute the BTF for each camera pair before installing them for surveillance. They claim that once this mapping is calculated then object correspondence is reduced to transformed histograms. But this assumption is not sufficient, because these cameras might be installed in a regions having different illumination than the color calibrated environment. Javed et al [9] propose a subspace based color brightness transfer function. They use a probabilistic PCA for object matching. They combine the color information with object motion and show the superiority of the combined approach. But in our experiments, we don’t have the information of camera position and many entry points are possible. Concepts of cumulative brightness transfer function (CBTF) and mean brightness transfer function (MBTF) are discussed in details in [10], [11]. Prosser et al [10] show in their experiments that CBTF is better than MBTF which seems to be contradicted the results of D’orazio et al [11]. In our experiments explain in section IV we find that MBTF and CBTF result are not significantly different.

In section II, we presents in details the recognition features and a method of object identification and tracking. In section
III, we will discuss the two existing techniques for inter-camera color calibration and propose a modification in the CBTF technique. In section IV, we discuss the results of our algorithm and compare it with other existing techniques in terms of ROC curve between precision and recall performance of re-identification, with and without color calibration. Section V presents the conclusion and some future works.

II. OBJECT RECOGNITION AND TRACKING

In previous work [1], we introduced a 1-D color based appearance model which preserves objects spatio-color information. This feature allows recognizing an object that exits and re-enters in the field of view of different cameras. We use a main database to store all detected objects. Each object that enters for the first time in any of the cameras field of view is stored once in a database. The database stores only invariant features (1D appearance model) of each object which is useful for the object recognition process. Position \( P = [x, y] \) and velocity \( V = [v_x, v_y] \) of object are commonly used in a single camera or calibrated multi-cameras as motion features. These features helps to recognize similar objects in a scene if they are far or moving with different velocities from each other.

A. Object Vertical Feature

The idea of an 1-D appearance model is to describe each object by a vector representing its projection on the main axis (head-foot axis) of the object, which will be considered as a vertical in the article. Because, we mainly address person recognition in a video surveillance systems. This vector is determined with a predefined height \( h \) and each value of \( y (y=1...h) \) is represented by the most representative color of the horizontal projection of this specified slice of the object.

**Vertical feature computation algorithm**

1) For each object fix the height value \( h \), where \( h \) is a constant number .
2) Extract segmented object (person) from image.
3) Rescale the object’s height \( h' \) to \( h \); if \( h' < h \) then use bicubic interpolation else use Gaussian kernel.
4) For each horizontal slice at level \( y_0 \) do:

5) Find the most representative color slice \( y_0 \).

\[
VF(y_0) = \xi(I(x, y_0))
\]

VF is a vector consisting of \( h \) elements for each color channel. Possible choices for \( \xi(I(x, y_0)) \) are mean and median color value of each color channel taken individually, or a particular color value minimizing a given criterion such as the quadratic sum of the distances in RGB color space. We performed experiments on \( \xi(I(x, y_0)) \) and found that mean and median color value give the best performance. Sometimes, we have the risk to lose the true object color if the horizontal projections of the object at some height slice have two or more very different colors. We also tried to use texture information, the results are not significantly improved due to poor quality of surveillance videos which don’t contain adequate level of detail. Fig. 1.a illustrates how large or small object are resized or re sampled to the predefined height \( h \). Fig. 1.b shows some sample objects and their VF representation. VF preserves some vertical texture information, face and hair color. VF technique is also computationally efficient and needs low memory storage.

B. Object position prediction

Object position prediction helps to estimate future objects position in the video frames and also can give some object-object occlusion prediction [1]. We use Kalman filter [12] to calculate the probabilistic position and to detect occlusion of objects in next frame. The Newtonian equation of motion allows to find the object position in the next frame. We model our states as a 4-D vector by using the object position \( [x^i, y^i] \) and its velocity \( [v_{x^i}, v_{y^i}] \) at discrete time \( t \) by \( X_t = [x^i_t, y^i_t, v_{x^i_t}, v_{y^i_t}] \). Dynamic equations of the system are described by the following non-linear equations:

\[
X_{t+1} = \begin{bmatrix}
x_{t+1}^i + v_{x^i_t} \Delta t + \frac{1}{2} a_{x^i_t} \Delta t^2 \\
y_{t+1}^i + v_{y^i_t} \Delta t + \frac{1}{2} a_{y^i_t} \Delta t^2 \\
v_{x^i_t} + a_{x^i_t} \Delta t \\
v_{y^i_t} + a_{y^i_t} \Delta t
\end{bmatrix}
\]

The non-linearity of object motion, the acceleration \( (a_{x^i_t}, a_{y^i_t}) \), is modeled by a process noise \( W_t \). \( \Delta t \) is the time difference between two consecutive frames. The matrix form of (2) is:

\[
X_{t+1} = A_{t+1,t} X_t + W_t
\]

where \( A_{t+1,t} \) is the state transition matrix which links next position to current object position. For more details see Bishop et al. [13].

C. Feature matching and object tracking

In order to match objects features, we use the Mahalanobis distance. It differs from the euclidean distance by its scale invariance and it also takes into account the objects color

![Figure 1. Sample objects, their VF representation and scaling](image)
variance in the database. Mahalanobis formula for VF is represented as

\[ D(VF^1, VF^2) = \sqrt{\sum_{y=1}^{n} \sum_{c=1}^{3} \left( \frac{V F^1(y, c) - V F^2(y, c)}{\sigma_{y,c}} \right)^2} \]  
(4)

\( \sigma_{y,c}^2 \) is the color variance of each element of vector VF in the database containing T objects.

Here we consider two cases. In a first case, we suppose that the object exits from one side and enters in a camera FOV from some other side. If \( D^O(O^c, O^n) > D_{ref} \) then a new object is created in the database. Selection of \( D_{ref} \) is important as smaller value of \( D_{ref} \) increase false negatives and higher value of \( D_{ref} \) increase false positives (see the section IV).

III. INTER-CAMERA COLOR CALIBRATION

Color calibration in non-overlapping environments is necessary as our recognition feature is mainly based on a color appearance model. Object appearance in different cameras may be very different even in controlled environment (indoor regions). Some important parameters influencing object’s color appearance are camera gain, focal length, aperture size, illumination conditions and scene geometry. Mapping function of one camera color space to a color space of another camera is called Brightness Transfer Function (BTF).

The two commonly used techniques are: mean brightness transfer function (MBTF) and cumulative brightness transfer function (CBTF). Brightness transfer function between two cameras is calculated during a training phase. During this stage we assume that we know the objects correspondence from \( C^i \) to \( C^j \). Let us assume object \( O \) enters in the camera \( C^i \) and after exiting \( C^i \) the same object enter in the camera \( C^j \). Normalized histograms for each channel is calculated for the two cameras and a BTF \( f_{i,j} \) between the cameras is computed from the two cumulative histograms. In the following subsections for simplicity we use histograms instead of RGB histograms and we use the same algorithm for calculating each color channel of objects. The algorithm for finding the BTF is explained in steps 1-3 as following:

BTF algorithm

1) Compute the normalized histograms \( I'(B_m) \) and \( I'(B_m) \) for each object of camera \( C^i \) and \( C^j \), where \( B_1,...,B_{256} \) are brightness values of each of histogram bins.

2) Find the cumulative histogram \( H_i(B_m) \) and \( H_j(B_m) \) as :

\[ H(B_m) = \sum_{k=1}^{m} I(B_k) \]  
(7)

3) \( H_i(B_m) \) and \( H_j(B_m) \) are normalized cumulative histograms for the same object \( O \). \( H_i(B_m) = H_j(f_{i,j}(B_m)) \) and the BTF mapping function is

\[ f_{i,j}(B_m) = H_j^{-1}(H_i(B_m)) \]  
(8)

In the following subsections, we will explain the method to calculate BTF \( f_{i,j} \) for MBTF, CBTF and our modified CBTF (MCBTF).

A. Mean Brightness Transfer Function

In mean brightness transfer function (MBTF) the \( f_{i,j} \) between two cameras \( C^i \) and \( C^j \) is obtained by calculating BTF for each object in camera pair \( C^i-C^j \). After calculating BTF for all objects images, the mean BTF is calculated from these BTF. The steps of MBTF are presented bellow:

1) Do the steps 1-3 explained above for each object appearing in camera pair \( C^i-C^j \).

2) Find the BTF for each of training objects pair.

3) Find the MBTF by taking the average of all the BTF. In general one object has only a limited number of colors, that is why each BTF curve gives better responses for the histogram regions where it is representative of objects colors. When we take the mean of all the BTF then the histogram regions having more colors represent a true correspondence for camera pair \( C^i-C^j \) but other regions lose informations due to the averaging process.

B. Cumulative Brightness Transfer Function

MBTF loses inter-camera color informations which are not present in majority of objects. This increases the problem
of object re-identification when the object exits from one camera and enters the field of view of another camera. If BTF is not properly calculated, then the object color mapping from one camera to the other one might be incorrect. This becomes the reason of false object identification. To overcome this problem, the cumulative brightness function is proposed in [10]. The algorithm is explained below:

1) Compute the histograms $h_1^i..h_i^k$ and $h_1^j..h_j^k$ for k objects per camera $C^i$, $C^j$.

2) Accumulate the brightness values of the histograms $h_1^i..h_i^k$ for camera $C^i$ and the histograms $h_1^j..h_j^k$ for camera $C^j$ and compute the cumulative histograms $H_i(B_m)$ and $H_j(B_m)$ as:

$$H(B_m) = \sum_{k=1}^{m} \sum_{i=1}^{k} h_i^k(B_k)$$

(9)

3) Normalize the cumulative histogram $H_i(B_m)$ and $H_j(B_m)$ by the total numbers of pixels in training set for $C^i$ and $C^j$.

4) Calculate BTF by using equation (8).

C. Modified Cumulative Brightness Transfer Function

Cumulative histogram technique explained in [10] claims that it solves the problem of MBTF which lose color information that are not present in the majority of objects. But in CBTF algorithm, it is clear from step 2 and 3 of subsection III-B that they accumulate all the histograms for the camera $C^i$ and $C^j$ respectively and then normalize the cumulative histograms for each camera. But in many cases, objects histogram do not have contributions to all the histogram bins. That is why if we take sum of the $B_m$ for all histograms and normalize its value by the total number of pixels in the training set, then a lot of information is lost. We propose to modify the steps 1, 2 and 3 of CBTF to improve its performance. The steps of our MCBTF are presented bellow:

1) Compute the histograms $h_1^i..h_i^k$ and $h_1^j..h_j^k$ for k objects per camera $C^i$, $C^j$

2) Find the mean value of $B_m$ of only those histograms that have a significant contribution to $B_m$. This is done as follows:

a) Find the normalized mean histogram $\overline{H_i}$ and $\overline{H_j}$ for camera $C^i$ and $C^j$ by using (10):

$$K_i^m = \{ k \in I; h_i^k(B_m) > \Theta \};$$

(10)

$I = [1..k]$ is the total set of histogram indices and $K_i^m$ is the subset of $I$ such that the selected histograms have a significant contribution to $B_m$. $m = [1..256]$ are total numbers of bins. $\Theta$ is basically minimum value of bin $B_m$ for being considered on its significance. The threshold $\Theta$ is selected in training period. Our experiments show that a typical value of $\Theta$ between 3 and 5 gives good results in most situations. The number of elements of $K_i^m$ will be noted as $\#K_i^m$

b) The average value of bin $B_m$ can then be computed as:

$$\overline{H_i}(B_m) = \frac{\sum_{k \in K_i^m} h_i^k(B_m)}{\#K_i^m}$$

(11)

3) Find the cumulative histogram $H_i(B_m)$ and $H_j(B_m)$ by using (7)

4) Calculate BTF by using (8).

IV. EXPERIMENTAL RESULTS

We have performed two series of experiments. We will first discuss the object tracking performance of our experiments in details. We will then explain the results of object re-identification, with and without using color calibration techniques in multi camera environments.

We compare our algorithm with two other methods: the first one is based on object motion estimated by Kalman filter and the second one is appearance and color based mean shift algorithm. We evaluate the tracking results on the well known PETS$^1$, CAVIAR$^2$ and VISOR$^3$ benchmark datasets. We have taken 7410 frames from 15 videos having at least one object present. These videos have many challenges like shadows, occlusions, irregular luminosity, object’s shape and size evolution, re-entrance of objects and low video quality. We compare each object’s position in}

1http://www.cvrg.rdg.ac.uk/PETS2009/a.html
2http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1
3http://www.openvisor.org/video_categories.asp
the video with a ground truth obtained by manual labeling of objects, and distinguish them for a visual control by drawing a rectangle around them using different colors.

Fig. 3 shows that we recognize objects in video sequences even with the change of view angle, similar clothes color and also under occlusion. Table I shows that Kalman filter gives good results on CAVIAR dataset due to less objects occlusion and interaction. Simple motion based technique can track the object successfully in this situation. Object color appearance base technique is unable to give good results due to color similarity between object-object and object-background. VISOR videos have small and large object sizes, similar objects, low and high video quality, uneven luminosity in video frames. Mean shift gives really poor results and Kalman filter fails to track objects due to occlusions. Kalman filter is unable to distinguish if the same or a different object enters in the scene as it only uses motion features. Our algorithm gives satisfactory results except for videos having a very low color quality.

PETS 2009 database have many challenges. There are many entry points, several occlusions of more than two persons. Many persons have similar clothes and they occlude each other. One person exits and an other person having the same clothes color enters at the same position in the scene. The same person exits from one side and enters from opposite side. Our algorithm is able to track the objects under above defined cases. In table I, the second last entry gives the recognition percentage of total objects present in all frames. Our algorithm success rate is 92%, Kalman has 82% and mean shift has last position having score 71%. Our proposed algorithm uses color, motion and spatial information that is why it supersedes both Kalman and mean shift. Last entry of table I is tracking rate (TR). We track 15-20 objects in images sequences on a laptop having a Core Duo processor (T2350) with 1.86 GHz clockrate, where as video frames size is 320 x 240. Our algorithm is faster than mean shift but slower than plain Kalman filter because Kalman filter is also integrated in our algorithm.

In the second series of experiments we have installed three cameras of different models (Fuji, Sony and JVC) with no overlapping field of view. We set camera C1 as reference and others cameras C2 and C3 are calibrated by finding their BTF $f_{2,1}$ and $f_{3,1}$, by using the three methodologies explained in section III. In the training phase we use 10 images/object for five objects present in the sequences for each camera. Illumination conditions and colors are different for these cameras. We plot the BTF of the C1-C2 pair of cameras for R, G and B channels. Fig. 2 shows that our MCBTF works better and it maps low intensity pixels to higher intensities pixels after intensity value 100 for camera C2 which has less light intensity. This is the point where our modification works better than CBTF and MBTF as in our experiments many persons have dark clothes. Histograms of every objects do not have contributions to all the bins. If we take the same method of CBTF, then BTF for pixels having intensities higher than 100 are mapped to lower values - which decreases the performance of object matching like CBTF and MBTF.

In second step, we allowed 12 objects to enter and exit from FOV of one camera to another camera more than 30 times from different direction in random order. In Fig. 4 objects are passing through in FOV of cameras C1, C2 and C3 in different time sequences and our algorithm draws the bounding box around these objects which remains constant in all camera even with the change of size, camera view angle, illumination changes and different color appearance.

We do object re-identification when they enter in the field of view of another camera. We plot ROC curve between precision (PR) and Recall (RE) for a set of values of $D_{ref}$. We fix the value of $\alpha = 0.97$ and $D_{th} = 60$ pixels and we change the value of $D_{ref}$ from 8 to 12. PR and RE of object re-identification for each algorithm is calculated by using equations (12) and (13).

\[
PR = \frac{TP}{TP + FP} \quad (12)
\]

\[
RE = \frac{TP}{TP + FN} \quad (13)
\]

If an old label is correctly assigned to an object, or created for a new object, then it is considered as a true positive (TP). If an old label is assigned to a wrong object, then it is

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CAVIAR</td>
<td>93.27%</td>
<td>85.64%</td>
<td>96.72%</td>
</tr>
<tr>
<td>VISOR</td>
<td>81.27%</td>
<td>64.70%</td>
<td>89.02%</td>
</tr>
<tr>
<td>PETS</td>
<td>73.18%</td>
<td>67.04%</td>
<td>91.355%</td>
</tr>
<tr>
<td>Overall</td>
<td>81.57%</td>
<td>70.86%</td>
<td>91.97%</td>
</tr>
<tr>
<td>TR (fps)</td>
<td>85.63%</td>
<td>7.18</td>
<td>39.32</td>
</tr>
</tbody>
</table>
Fig. 5 illustrates that our modification to CBTF increases significantly the object re-identification performance, both in terms of precision and of recall. MCBTF has higher value of precision and recall than all the three other techniques. In ROC curves, performance of CBTF and MBTF are comparable, without clear advantage of one approach in front of the other. This corroborates the contrasted results presented by [10] and [11]. The three color calibration techniques show nevertheless a slight improvement as compared to the basic approach that does not calibrate cameras (WCC). This latter method produces more FP specially in camera 2 which is installed in the region having lower illumination. Hence its re-identification results are worse than any of the techniques that use color calibration. We can then conclude from our object re-identification results that MCBTF gives re-identification performance significantly better than CBTF and MBTF.

V. CONCLUSION AND FUTURE WORKS

We extended our simple 1-D appearance model for object tracking and re-identification in non-overlapping multi-camera video surveillance system. We proposed the modification to the well known CBTF color calibration technique and showed a significant improvement in tracking and re-identification. In general, the algorithm gives satisfactory results, but there are situations in which it fails: if object’s height is smaller than 30 pixels or in the case of very low video quality. Modified cumulative brightness technique works satisfactory even for a new objects which have not been used during the training time. In future works, we are planning to propose BTF updating strategies, in order to cancel out the effects of changes in illumination conditions after the training time.

REFERENCES


