A Fast Non-Overlapping Multi-Camera People Re-Identification Algorithm and Tracking Based on Visual Channel Model

Chi-Chia SUN††, Nonmember, Ming-Hwa SHEU††, Member, Jui-Yang CHI††, and Yan-Kai HUANG††, Nonmembers

SUMMARY In this paper, a nonoverlapping multi-camera and people re-identification algorithm is proposed. It applies inflated major color features for re-identification to reduce computation time. The inflated major color features can dramatically improve efficiency while retaining high accuracy of object re-identification. The proposed method is evaluated over a wide range of experimental databases. The accuracy attains upwards of 40.7% in Rank 1 and 84% in Rank 10 on average, while it obtains three to 15 times faster than algorithms reported in the literature. The proposed algorithm has been implemented on a SOC-FPGA platform to reach 50 FPS against low resolution, occlusions, and pose. In [5], metric learning methods and an adaptive threshold method are applied to avoid insufficient and sub-optimal conditions for the verification problem for a fixed threshold. In [10], metric learning is used for people re-identification and, then, to further estimate scaling for distances in a vector space high-dimensional features are extracted from the source color image. However, there is spatial misalignment between the patterns of the same object in multi-cameras. To handle spatial misalignment, [11] referred to the use of mid-level filters to find the patch and use the patch to calculate the distance between two objects. Paisitkriangkrai et al. [12] proposed a metric learning and ranking algorithm for people re-identification. The basic idea behind metric learning is to find a mapping function from the feature space to the distance space with certain merits, such as feature vectors from the same person being closer than those from different ones. In [13], it leveraged low-level feature descriptors to approximate the appearance variants in order to discriminate individuals by using a sparse linear reconstruction model. Liao et al. [14] proposed an effective feature representation called Local Maximal Occurrence (LOMO) to maintain a stable representation against viewpoint changes and also proposed a subspace and metric learning method called Cross-view Quadratic Discriminant Analysis. Wang et al. [15] proposed using spatial-temporal descriptors to re-identify pedestrians. Its features include HOG3D and the gait energy image. By designing a flow energy profile, walking cycles are detected so that frames around the local minimum/maximum are used to extract motion features. Finally, reliable spatial-temporal features are selected and matched through a discriminative video ranking model. Sun et al. [16] proposed an efficient people re-identification method with latent variables, which represents a pedestrian as the mixture of a holistic model and three flexible models, including vertical misalignment, horizontal misalignment, and leg posture variations.

1. Introduction

As processor speeds become faster and the cost of the hardware is reduced, there is an ever increasing benefit to the important field of computer vision. Computer vision can be used in many areas such as production line inspection, pedestrian identification system, and vehicle statistics on road. In the past, these devices required a significant effort to achieve the desired results, so an automated system was gradually used. There are already many applications in security monitoring, such as image monitoring and identification systems. Hence, image processing systems have enormous future potential, and intelligent monitoring systems will be used in machine vision and surveillance systems. People re-identification from a different viewpoint is a challenging task, including viewpoint change and occlusions, cluttered background, and lighting changes. In general, re-identification methods can be mainly divided into two categories: feature design and selection [1]–[4] or vector learning [5]–[9].

In [2], it used pedestrian symmetry and asymmetry in combination with three complementary modes: the color histogram; the spatial arrangement of colors into stable regions; and the presence of recurrent local motifs with high entropy. The approaches are applied to solve the robustness against low resolution, occlusions, and pose. However, deep learning requires a lot of training and different training materials, which may result in uncertain effects in practical applications [17], [18]. Contrary, the traditional algorithm does not have the above problems and, meanwhile, is easier for hardware implementations compared with deeper learning methods.
In this paper, a new nonoverlapping multicamera person re-identification algorithm is proposed. First, it uses color correction for a multicamera to reduce the effect of chromatic aberration and different light sources between different scenes. Second, inflated major color features for re-identification are applied to reduce the computation complexity while still retaining high accuracy of object identification. Compared with the most recent works, the accuracy of Rank5 and Rank10 is 9% higher than others on average, and Rank 1 has only a 0.4% difference. However, because the proposed algorithm is based on simple characteristics of the inflated major color, the computation time is dramatically improved, with three to 15 times faster speed than algorithms reported in the literature. The proposed algorithm has been implemented on a low-cost SOC-FPGA embedded platform for real-time processing, without any coding optimization, and it can easily reach 50 FPS with 1280 x 720 HD resolution and 25 FPS with 1920 x 1080 FHD resolution. The advantages of the presented architecture are (1) the accuracy attains upward of 40.7% in Rank 1 and 84% in Rank 10 on average compared with other algorithms reported in the literature; (2) it requires much less computation complexity among the other approaches but only has 0.4% difference in Rank 1; (3) it is worth noting that the proposed fast algorithm is ideal for those low power embedded devices that must solve person re-identification within multiple camera scenarios.

This paper is organized as follows. Section 2 provides a brief description of the proposed person re-identification algorithm. Section 3 defines the cross-video matching method and object tracking algorithm. The performance is analyzed in Sect. 4, and Sect. 5 concludes this paper.

2. Proposed Algorithm

Figure 1 illustrates the flow chart of feature extraction for person re-identification based on visual channel information. The algorithm first detects the moving object from the input video stream, labels the detected object, and then segments the objects. Next, cross-video object characterization is performed based on a visual channel model; the maximum, minimum, and saturation of each pixel are calculated, and, if the saturation is greater than the threshold, then it outputs the cone channel pixel; otherwise, it outputs the rod channel. Then, the images of the cone channel and the rod channel are morphologically inflated and clustered. Finally, the color, area, and position information for each block are recorded as the characteristic for object rematching.

2.1 Moving Object Detection

ViBe, a universal background subtraction algorithm for video sequences proposed by Barnich et al. [19], is a pixel-level background model, the major feature of which is the background model update by random selection of pixel samples needing replacement, and random selection of neighborhood pixels for updating. As the uncertainty in a cross-video tracking system cannot be determined, a random update strategy is chosen in our proposed system with a confidence to overcome the uncertainty of pixel change, such as influence of light change, and viewpoint changes between different cameras. Equation (1) shows the establishment of a sample background model for each pixel. Equation (2) calculates the classification of pixels and background model similarity, if similar, then classifies it as background, as shown in Fig. 2.

\[ M(x) = \{v_1, v_2, \ldots, v_N\} \]  
\[ S_R(v(x)) \cap \{v_1, v_2, \ldots, v_N\} \geq Th. \]  

where \( M(x) \) is the sample background model; \( x \) is the input pixel; \( v_i \) is the \( i_{th} \) sample value in the background sample set; \( N \) is the number of samples, here \( N \) is set to 20; \( S_R(v(x)) \) is a sphere centered on \( v(x) \), \( R \) is the distance to determine the threshold, which is set to 20; and \( Th \) is the threshold, which is set to 2. 

\[ M(x) = \{v_1, v_2, \ldots, v_N\} \]  
\[ S_R(v(x)) \cap \{v_1, v_2, \ldots, v_N\} \geq Th. \]
2.2 Object Segmentation and Visual Channel Model

The object shapes detected by ViBe background model might be broken or jagged. Therefore, we use Sobel edge detection to complete the object segmentation; thus the neighborhood area of the object is clustered together. Figure 3 shows the results of the object segmentation, where similar pixels will be connected.

Next, a Visual Channel Model to imitate human visual cells will be created for cross-video object characterization. First, it determines the maximum, minimum, and saturation of each pixel, followed by imitating the visual cells by defining a saturation threshold. When the pixel is brighter, this means that is color is more important; otherwise, the color is not important. \( S_{th} = \frac{1}{P_{max}} \), where \( P_{max} \) is the maximal value of the input pixels. Then, we will use the saturation threshold to transform the segmented object image with several sub-blocks into a cone channel with color information or a rod channel, which does not have color information, as mentioned in Eq. (3).

\[
\begin{align*}
\text{Cone}_CH(x,y) &= \begin{cases} 
(M(x,y), \text{Mid}(x,y)), & \text{if } \text{Sat} \geq S_{th} \\
0, & \text{otherwise}
\end{cases} \\
\text{Rod}_CH(x,y) &= \begin{cases} 
\text{GrayLevel}_{x,y}, & \text{if } \text{Sat} \geq S_{th} \\
0, & \text{otherwise}
\end{cases}
\end{align*}
\]

\[\begin{align*}
\text{Max}(x,y) &= \text{Max}(R(x,y), G(x,y), B(x,y)) \\
\text{Min}(x,y) &= \text{Min}(R(x,y), G(x,y), B(x,y)) \\
\text{Saturation}(x,y) &= 1 - \frac{\text{Min}(x,y)}{\text{Max}(x,y)}.
\end{align*}\tag{3}\]

where \( \text{Max}() \) is the largest channel; \( \text{Mid}() \) is the middle of the channel. Figure 4 shows a characterization result based on the proposed visual channel model.

After the visual channel image transformation, we divide the channel image into two layers: the cone channel and the rod channel. To improve the re-identification accuracy, a morphological expansion method is applied to visual channels. The similar color group is clustered together in conjunction with the original channel image and the expanded channel image to find a large area and label re-markings as described in Eq. (4).

\[
\begin{align*}
I_{\text{cluster}} &= \begin{cases} 
\text{cluster}, & \text{if } I_{\text{VC}} \cap I_{\text{VC},\text{Dil}} \\
0, & \text{otherwise}
\end{cases} \\
\text{VC} &\in \text{Cone}_CH, \text{Rod}_CH
\end{align*}\tag{4}
\]

where \( I_{\text{cluster}} \) is the result of cluster color, \( \text{cluster} \) is clustered color; \( I_{\text{VC}} \) is visual channel image from Eq. (3), and \( I_{\text{VC},\text{Dil}} \) is the visual channel after expansion as shown in Eq. (5). Next, the object characterization will be extracted, it will record the color information, area ratio, and location ratio, as shown in Fig. 5 (b).

\[
\begin{align*}
I_{\text{VC},\text{Dil(near)}} &= \begin{cases} 
I_{\text{VC}}(x,y), & \text{if } I_{\text{VC}} \geq 0 \\
0, & \text{otherwise}
\end{cases} \\
\text{near} &\in (x,y-2), (x-1,y-1), (x,y-1), \\
&\quad (x+1,y-1), (x-2,y), (x-1,y), (x,y), \\
&\quad (x+1,y), (x+2,y), (x,y+2), (x-1,y+1), \\
&\quad (x,y+1), (x,y+1), (x+1,y+1).
\end{align*}\tag{5}
\]
3. Match Across the Camera

Figure 7 shows the system flow chart of the proposed non-overlapping multi-camera people tracking and re-identification algorithm. At the beginning of initialization, color calibration between cameras to reduce the effect of chromatic aberration will be processed in order to avoid the influences from light, type of camera and location [20]. After color calibration, the ViBe background model [19] is created, followed by moving object segmentation and cross-video object characterization based on the proposed algorithm and then tracking of selected objects based on the tracking algorithm in [21]. At the beginning of the tracking algorithm, the Region Of Interest (ROI) is selected to store the target’s position and image information. Then, the edge-detection algorithm is used to find the edge of tracking object, and then the edge will be individualized in order to reduce the computational complexity of the tracking algorithm as the tracking target feature. Figure 6 shows the tracking feature extraction and characterization. Then, the feature points of the template are compared with the feature points of the search range to find the highest similarly matching points. The ROI of the tracking target is determined by an image surrounded by a tracking block of size $N \times N$, and the central coordinate of the square window. There are many methods for characterization, such as gradients, colors, or textures. Here we use the adjacent texture cluster characterization method. Further detailed information of object tracking can be found in [21].

Next, we determine whether all objects have been tracked, to determine whether or not an object disappears or a new one is added. If an object disappeared, the crossed-video system will record the disappeared object features and then send it to the adjacent camera’s devices via internet; if there is a new object added, it will try to match the objects from the adjacent camera’s devices for cross matching. If successfully matched, it will indicate which camera the object came from and show the matching result; otherwise, the object is considered as appearing for the first time in the system.

Equation (6) represents the matching method of objects across the cameras. It can be divided into appearance correlation and location relevance in Eq. (7). The position correlation is established based on the camera topology map. Assuming camera $C_1, C_2, \ldots, C_N$, the location of the camera will use the relative position of the camera to establish a topology map to describe the relevance of a camera.

$$d(O_i, O_n) = \begin{cases} d_{VC}(VC(O_i), VC(O_n)) & \text{if}(CT_{i,n} = 1) \\ \infty & \text{otherwise} \end{cases}$$

$$d_{VC}(VC(O_i), VC(O_n)) = \alpha \times d_{color}(O_i, O_n) + \beta \times d_{loc}(O_i, O_n) + \gamma \times d_{area}(O_i, O_n).$$

where $d_{color}$ represents the color difference of H, S, and V between two objects. $d_{loc}$ stands for location difference of left, right, up and down positions between two objects, and $d_{area}$ is the difference of pixel numbers between two objects. Here, $\alpha$ is set to 0.55, $\beta$ is set to 0.3, $\gamma$ is set to 0.15 through numerous simulation based on VIPeR Rank1 datasets [1] as shown in Table 1.

<table>
<thead>
<tr>
<th>Parameter of camera topology.</th>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>$\gamma$</th>
<th>VIPeR Rank1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha=0.33$</td>
<td>$\beta=0.33$</td>
<td>$\gamma=0.33$</td>
<td></td>
<td>17.2%</td>
</tr>
<tr>
<td>$\alpha=0.55$</td>
<td>$\beta=0.15$</td>
<td>$\gamma=0.15$</td>
<td></td>
<td>38.5%</td>
</tr>
<tr>
<td>$\alpha=0.55$</td>
<td>$\beta=0.15$</td>
<td>$\gamma=0.3$</td>
<td></td>
<td>28.3%</td>
</tr>
<tr>
<td>$\alpha=0.3$</td>
<td>$\beta=0.55$</td>
<td>$\gamma=0.15$</td>
<td></td>
<td>12.6%</td>
</tr>
<tr>
<td>$\alpha=0.15$</td>
<td>$\beta=0.55$</td>
<td>$\gamma=0.3$</td>
<td></td>
<td>9.4%</td>
</tr>
<tr>
<td>$\alpha=0.15$</td>
<td>$\beta=0.3$</td>
<td>$\gamma=0.55$</td>
<td></td>
<td>7.5%</td>
</tr>
<tr>
<td>$\alpha=0.3$</td>
<td>$\beta=0.15$</td>
<td>$\gamma=0.55$</td>
<td></td>
<td>9.2%</td>
</tr>
</tbody>
</table>
Table 2  Comparisons between the proposed algorithm and other recent works on average Rank1, Rank5 and Rank10 accuracy.

<table>
<thead>
<tr>
<th></th>
<th>VIPeR dataset</th>
<th>iLIDS dataset</th>
<th>CUHK01 dataset</th>
<th>PRID450S dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R-1</td>
<td>R-5</td>
<td>R-10</td>
<td>R-1</td>
</tr>
<tr>
<td>2012 [9]</td>
<td>19.3%</td>
<td>48.9%</td>
<td>64.9%</td>
<td>20.6%</td>
</tr>
<tr>
<td>2013 [5]</td>
<td>29.3%</td>
<td>61.0%</td>
<td>76.0%</td>
<td>21.1%</td>
</tr>
<tr>
<td>2015 [14]</td>
<td>40.0%</td>
<td>68.1%</td>
<td>80.5%</td>
<td>43.0%</td>
</tr>
<tr>
<td>2017 [16]</td>
<td>39.6%</td>
<td>72.0%</td>
<td>84.5%</td>
<td>41.7%</td>
</tr>
<tr>
<td>Ours</td>
<td>39.2%</td>
<td>72.1%</td>
<td>84.9%</td>
<td>42.3%</td>
</tr>
</tbody>
</table>

The multi-camera topology is defined as:

\[
CT_{n,m} = \{(\theta_{n,m}, D_{n,m})|\text{Associate}(C_n, C_m) = True\}
for n, m = 1, 2, \ldots, N.
\] (8)

where \(CT_{n,m}\) is the topological value between \(C_n\) and \(C_m\) cameras, \(\theta_{n,m}\) is the angle of the two cameras, and \(D_{n,m}\) is the distance between \(C_n\) to \(C_m\); \(\text{Associate}(C_n, C_m)\) represents the neighbor relation for two camera \(C_n\) and \(C_m\) based on the topology. If \(\text{Associate}(C_n, C_m)\) is true, it means that there is a walking path between \(C_n\) and \(C_m\).

4. Experimental Results

For comparison, we followed the evaluation scheme described in [14]. For different datasets, we evenly distribute the target and test samples. Camera A stetted as the main camera, and camera B stetted as a neighbor node from the main camera. We match each of the main camera images with one of the cameras from the camera. Each experiment of the dataset was repeated 10 times to overcome the effects of randomness. Four public datasets are evaluated i.e., VIPeR [1], iLIDS [22], CUHK01 [8], and PRID450S [23]. The VIPeR dataset contains 632 pedestrians captured outdoors and therefore has complex lighting changes. In addition to changes in lighting, it also contains a considerable number of viewing angles, where the change between two cameras is usually larger than 90 degrees. The iLIDS dataset contains 119 pedestrians captured in two to eight cameras at an airport for a considerable space asymmetry, which is challenging. VIPeR and iLIDS datasets are the most often compared ones as shown in Fig. 8. In additional, the CUHK01 person re-identification dataset uses a single camera view as the probe set and the other as the gallery set. There are 971 identities, 3884 images, manually cropped. PRID450S dataset was based on the PRID 2011 dataset and contains 450 image pairs recorded from two different, static surveillance cameras.

Table 2 and shows the Rank1, Rank5 and Rank10 results of the proposed algorithm. In Fig. 9, the proposed algorithm recorded the color of the clothes, pants, and backpack straps and perfectly matched them to the clothes, pants and backpack in another camera. In the middle, because of the relationship between the visual channel, it can complete the re-identification when the target moves from the bright environment into a shadow. Furthermore, the camera is rotated by 180 degrees over the target, the proposed algorithm still completed person re-identification successfully.

We also compared the results to recent object re-identification algorithms for the performance comparison. Table 2 show the simulation time in Matlab (CPU i7-4790 3.6GHz, 8GB RAM, and Win 7). Compared with [14], the average accuracy of the Rank5 and Rank10 are 1.4% and 2.65% higher than the most recent research, Rank 1 has only a 2.05% difference. However, because the proposed algorithm is based on the simple characteristic of major color, the computation time is three to 15 times faster than what is reported literature. Compared with [14], the proposed algorithm is 8.4× faster, and accuracy is almost the same. Compared with the latest work [16], it is at least 15× faster while providing an improved detection rate by 3.07%.

In addition to the software simulation, we have implemented the proposed algorithm on a low-cost ZYNQ compatible SoC-FPGA embedded platform without coding op-
Table 3  Comparisons of the computation speed between the proposed algorithm and other recent works in Matlab.

<table>
<thead>
<tr>
<th>Method</th>
<th>Frame per Second</th>
</tr>
</thead>
<tbody>
<tr>
<td>2012 [9]</td>
<td>2.6</td>
</tr>
<tr>
<td>2013 [5]</td>
<td>1.8</td>
</tr>
<tr>
<td>2015 [14]</td>
<td>0.9</td>
</tr>
<tr>
<td>2017 [16]</td>
<td>≪ 0.5</td>
</tr>
<tr>
<td>Ours</td>
<td>7.6</td>
</tr>
</tbody>
</table>

Fig. 10  The system flowchart of algorithm on a SoC FPGA heterogeneous computing platform.

Fig. 11  The result of the proposed algorithm implemented on SoC FPGA ARM embedded platform.

timization (Dual A9 900Mhz and 512MB DDR-3) as shown in Fig. 10. Object tracking is realized in FPGA acceleration with high segmentation and tracking efficiency. Because the computation overhead of our proposed fast person re-identification algorithm based on visual channel model is light, a single-core Corext ARM A9 processor can easily handle the re-identification algorithm. Two surveillance cameras using a BNC connector is controlled by an ARM processor through the AXI4 channel; next, the YCbCr color space transform and tracking results will be transformed to RGB and then stay at a reserved DMA memory space with Linux kernel. Figure 11 shows the implementation of the proposed algorithm on a SoC FPGA platform using FPGA heterogeneous computing acceleration. It can reach 50 FPS with 1280×720 HD resolution and 25 FPS with 1920×1080 FHD resolution.

5. Conclusion

In this paper, a new nonoverlapping multi-camera object tracking and people re-identification algorithm is proposed. In order to reach real-time computation with a multi-camera system, inflated major color features for re-identification are applied. The proposed algorithm accuracy attains upwards of 49.7% in Rank 1 and 87.55% in Rank 10 on average. The proposed algorithm has also been implemented on a low-cost SoC-FPGA embedded platform, and can reach 25 FPS with 1080P resolution. Notably, the proposed fast algorithm is ideal for low-power embedded devices that must solve multiple object tracking problems or be employed in security applications.

Acknowledgments

This research was supported in part by the Ministry of Science and Technology of Taiwan, under the contracts of MOST-105-2221-E-150-069 and MOST-105-2221-E-224-055-MY2.

References


Chi-Chia Sun

received his B.S. and M.S. degree in Computer Science and Engineering from National Taiwan Ocean University (2000-2004) and Electronic Engineering from National Taiwan University of Science & Technology (2004-2006) respectively. From April 2008 to March 2011, he worked as a research assistant at Dortmund University of Technology and received his doktor ingenieur degree with German Academic Exchange Service (DAAD) scholarship supports. Currently, he is an associate professor in the Department of Electrical Engineering of National Formosa University, Taiwan. His research interests are in Image Processing, System Integration and VLSIFPGA design. He is also a member of IEEE and TICD society.

Ming-Hwa Sheu

received the M.S. and Ph.D. degrees in Electrical Engineering from National Cheng Kung University, Tainan, Taiwan, in 1989 and 1993 respectively. He is presently a distinguish professor with the Department of Electronic Engineering, National Yunlin University of Science & Technology, Yunlin, Taiwan. He served as the chairman of the Department of Electronic Engineering from 2008 to 2011. His research interests include CAD/VLSI, digital signal process, algorithm analysis, and data communication.

Jui-Yang Chi

received his B.S. (2011-2015) and M.S. (2015-2017) degree in Electronic Engineering from National Yunlin University of Science & Technology respectively. His research interests are in Image Processing and FPGA Implementation.

Yan-Kai Huang

received his B.S. (2012-2016) and M.S. (2016-2018) degree in Electronic Engineering from National Yunlin University of Science & Technology respectively. His research interests are in Image Processing and FPGA Implementation.