I-012

Linking Trajectories of a Moving Object on Multiple Cameras with Different Angles and Locations

イ　グスティ　バグス　バスカラ　ヌグラハ† 野田 晋平† 森田 啓義†
I Gusti Bagus Baskara Nugraha Noda Shinpei Morita Hiroyoshi

1. Introduction

Nowadays, digital personal video recording devices are everywhere, such as in handycam, compact digital camera, or even in mobile phone. With the popularity increase of social networking sites, e.g., Youtube and Facebook, people do not hesitate to share their recorded video to family members, friends, or even the public through these sites. To facilitate this, some cameras are even equipped with wireless link so that the users can easily share or upload the video directly to any video sharing site. Therefore, abundant number of personal home videos can now be easily accessed by everyone over the Internet.

From those videos, it is possible that someone wants to search for redundant recorded video, which was taken at the same place in the same time, that feature the same person that he or she would like to see. For example, in the festivals or sports events, many people make a video recording of one or some interesting scenes so that their cameras will shoot at the same object, but from different position and angle.

We propose an unsupervised mechanism to correspond the trajectories of a person in multiple videos that are recorded at the same place in the same time by using different multiple cameras independently. The correspondence of the person can be analyzed by using the location and shooting direction information from camera. Modern cameras, including smartphones, have been equipped with Global Positioning System (GPS) and digital compass features so that the required information can be obtained from those features.

We assume that the path of the tracked person must be linear. The proposed object correspondence algorithm then estimates the direction of that path. Since every camera records the object from difference position and angle, the direction of the person seen by the camera is different. Hence, depending on the camera used as reference, the direction information from all cameras have to be calibrated. Based on the object's path and its calibrated direction information, the position of the object in the recorded image, and the estimated size of the object on the recorded image, we can link the trajectories among those videos and find the corresponding object in all other videos.

Unlike the other proposals, e.g., [1][2][3][4], our algorithm works on compressed domain since most digital video cameras give the output in compressed format, particularly H.264/AVC format. We use the algorithm proposed in [5] to detect and track moving objects and then link the trajectory created by the object on every camera.

This paper is organized as follows. Section 2 describes the previous work on this field. Section 3 explains our algorithm. Section 4 shows our experiment results, and Section 5 concludes our work.

2. Related Work

Researches on linking object on multi-camera have been reported in [1][2][3][4]. Javed et al. [1] estimate and learn the inter-camera travel time, which is a time interval when an object exiting from one camera and entering into another, and also the object's exit/entry locations during a training phase. By using these information and color histogram of the object, they establish correspondences by using maximum a posteriori (MAP). Similarly, Ikegami et al. [2] also use color information and Bayesian framework to establish object correspondence. Ukita et al. [3] use the objects' entry/exit time and Noda et al. [4] use clustering technique to make the correspondences.

All the above methods assumed that the people use a predefined path that they tend to follow, e.g., road or sidewalk. They also use training data in advance to estimate the objects' appearances.

In this paper we use a different approach as follows. First, an object may move from somewhere to any arbitrary direction without any path helper, e.g., road or pedestrian sidewalk. Second, the cameras are placed at low altitude positions since people tend to handle cameras at relatively low height. The previous works assume security cameras that are commonly placed at higher position, e.g., the building's ceiling.

However, we also make the following assumptions. First, the object's path must be linear so that the object should not change direction abruptly. Second, the camera's geographical position and also its shooting direction must also be recorded properly based on the GPS and compass data. Third, the cameras's shooting position is parallel to the ground.

This paper will only focus on correspondence algorithm. The object's position and its size are estimated by using object tracking algorithm proposed by Weng et al. [5]. It is a compressed domain tracking algorithm that detects moving object based on macroblock size and then estimates its path by using a modified Kalman filter algorithm. We use compressed domain approach since modern digital video cameras have already compressed the recorded video by using video compression standard, e.g., H.264/AVC. Processing video in compressed domain speeds up the computation time and enables real-time applications.

† The University of Electro-Communications

3. Proposed Method

Our proposed method consists of four sequential steps: estimation of the direction of moving object's path, quantization of the direction, direction calibration, and object correspondence (see Fig. 1). We will explain the method used in each step in the following subsections.

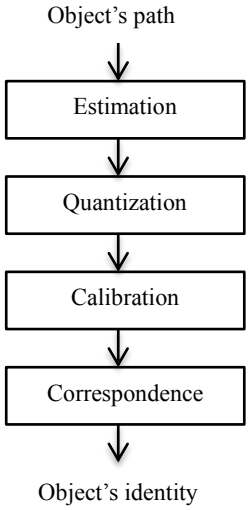


Fig. 1. Flowchart of the Proposed Algorithm

3.1 Estimation of the Direction of Object's Path

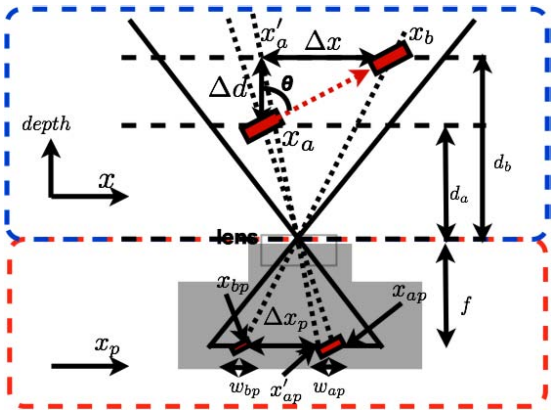


Fig. 2. Object's Trajectory and Its Image Projection

We estimate the direction of the trajectory of a moving object based on the size and the position of the object on the image plane as illustrated on Fig. 2. The figure depicts an object that crosses the camera's lens' field of view (FOV) linearly from position a to position b. The camera's lens captures the light from the object and projects it onto its imagery sensor. Based on the image data, we estimate the real direction of the object θ.

The value of θ is derived from Δd and Δx. We calculate Δd as follows. Let d_a and d_b denote the distance between the camera's lens plane and the object plane at position a and b, respectively, where

$$d_a = f\frac{\omega}{\omega_{ap}}$$
$$d_b = f\frac{\omega}{\omega_{bp}} \quad . \tag{1}$$

Then,

$$\Delta d = d_b - d_a = f.\omega.\frac{\omega_{ap}-\omega_{bp}}{\omega_{ap}\omega_{bp}}. \tag{2}$$

Similarly, Δx is calculated as follows:

$$\begin{aligned}\Delta x &= \Delta x_p.\frac{\omega}{\omega_{bp}} \\ &= \left(x_{bp} - x'_{ap}\right).\frac{\omega}{\omega_{bp}} \\ &= \left(x_{bp} - x_{ap}.\frac{d_b}{d_a}\right).\frac{\omega}{\omega_{bp}} \quad .\end{aligned} \tag{3}$$

Hence, θ can easily be obtained from the following equation:

$$\begin{aligned}\tan\theta &= \frac{\Delta x}{\Delta d} \\ &= (x_{bp} - x_{ap}.\frac{\omega_{ap}}{\omega_{bp}}).\frac{\omega_{ap}}{f.(\omega_{ap}-\omega_{bp})}\end{aligned} \tag{4}$$

3.2 Quantization

After obtaining $\theta(i)$ from the video taken by camera i, we have to quantize $\theta(i)$ into a discrete value to simplify the analysis. In our case, we divide $\theta(i)$ into eight possible values $q_\theta(i)$ as shown on Fig. 3, where

$$q_\theta(i) = \left\lfloor\frac{\theta(i)}{45}\right\rfloor. \tag{5}$$

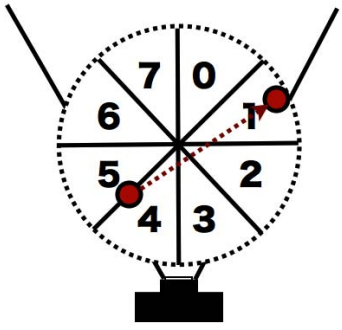


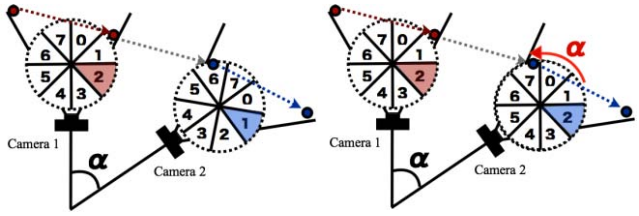
Fig. 3. Quantization of the Direction of Moving Object's Path

In the figure, we give an example of quantization of $\theta(i)$ into $q_\theta(i) = 1$.

3.3 Calibration

After quantizing $\theta(i)$ into $q_\theta(i)$, we need to calculate the relative direction of a moving object that is seen by a camera. For the same object, $q_\theta(i) \neq q_\theta(j)$ for $i \neq j$ because of the difference in the shooting angle. Based on the difference of shooting angle between cameras, when we use camera i as reference, $q_\theta(j)$ must be calibrated.

For example, Fig. 4 depicts two cameras whose shooting angles differ by α. Based on the figure, $q_\theta(1) = 2$ and $q_\theta(2) = 1$ (see Fig. 4(a)). When camera 1 is used as reference, $\theta(2)$ must be adjusted to a new value $\theta'(2)$ based on the value of α. Then, this value should be quantized so that we get a new value ${q'}_\theta(2) = 2$ (see Fig. 4(b)).



(a) (b)

Fig. 4. The value of $q_\theta(1)$ and $q_\theta(2)$ (a) before and (b) after calibration.

3.4 Correspondence

We use two parameters to link the object's trajectory on several video data recorded by multiple cameras: the difference value ΔT between the object's exit time T_{out} and entry time T_{in} on the camera's FOV, and the value of $q_\theta(i)$ and ${q'}_\theta(j)$. In other word,

$$\Delta T = T_{in} - T_{out} \quad (6)$$

$$q_\theta(i) = q'_\theta(j) \quad (7)$$

Based on Fig. 5, if $\Delta T < T_{th}$, where T_{th} is a predefined threshold, and $q_\theta(i) = q'_\theta(j)$, then we conclude that the two trajectory are actually come from the same object.

4. Experiment Results

We evaluate the accuracy of the proposed method on the estimation of the direction of moving object's trajectory in Subsection 4.1 and the accuracy of the linking method in Subsection 4.2.

4.1 The Accuracy of the Estimation of Direction

(1) Experiment Method

This experiment uses the iPhone 4's camera to record video. The camera's specification and experiment parameters are depicted on Table 1. During recording, our algorithm notices the moment where the object enters the camera's FOV (IN) and the moment where the object completely disappears from the camera's FOV (OUT), as shown on Fig. 6. The experiment is performed in a room where the lighting condition is constant. Fig. 7 shows the environment of the experiment, including the object used for the experiment.

The object moves across the camera's FOV at a constant speed from a randomly selected point towards several directions as mentioned in Table 1. Our task is to estimate the direction using the recorded video data. An Apple's MacBook, with a 2.6 GHz Intel Core Duo processor, runs our algorithm to process the video data.

Table 1. Experiment Parameters

	Experiment Setup
Camera type	iPhone 4's camera
Resolution	1280×720 pixel @30 fps
Focal length	34.65 mm
Pixel size	1.75 μm × 1.75 μm
Object's direction	-60°, -45°, -30°, 30°, 45°, 60°

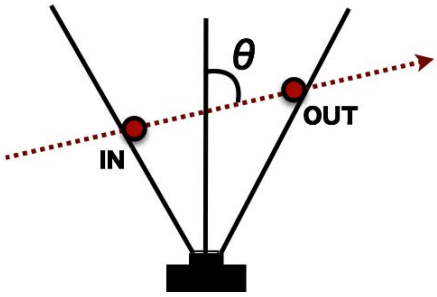


Fig. 6. The Object's Entry (IN) and Exit (OUT) Moments

Fig. 7. Environment for the Experiment

(2) Experiment Result

Table 2 shows the experiment result from three experiments for predefined direction's angles. The result shows that we can estimate the direction of a moving object with high accuracy, where the highest error value is about 10% (in case of 30°).

Table 2. The Accuracy of θ Estimation

	Predefined Direction					
	-60°	-45°	-30°	30°	45°	60°
First	-58.2	-42.4	-27.5	27.0	44.7	58.9
Second	-59.2	-44.0	-28.2	28.2	42.1	59.6
Third	-58.8	-43.4	-28.5	28.7	44.0	59.2

4.2 The Accuracy of The Correspondence Method

(1) Experiment Method

We perform the experiment on a real outdoor environment, where the objects are human (see Fig. 8). The figure shows blocks with various colors where each color denotes one of H.264/AVC macroblock (MB) partitions: 8×8, 8×16, 16×8, or 16×16 pixel partition. The object tracking algorithm proposed in [5] detects and tracks a moving object based on this

information. We set up two cameras with two placement patterns as shown on Fig. 9. For each pattern, an object moves across the cameras' FOVs at a constant speed without changing its direction.

The accuracy is measured by using two metrics called *precision* (P) and *recall* (R), which is defined as

$$P = \frac{CDC}{CC} \quad (8)$$

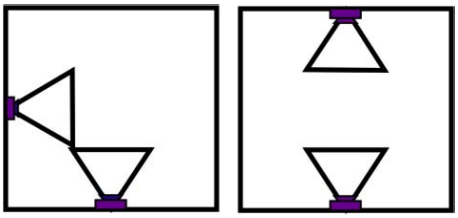
$$R = \frac{CDC}{DC}, \quad (9)$$

where CDC or *Correct Detect Count* is the number of correctly detected paths, CC or *Correct Count* is the actual number of paths, and DC or *Detect Count* is the number of detected paths.

(2) Experiment Result

Table 3 shows the experiment results for each pattern where the numbers of objects N are seven. The accuracy of the proposed method is above 70%

Fig. 8. Environment of the Experiment



(a) Pattern 1 (b) Pattern 2

Fig. 9. Camera's Placement

Table 3. The Accuracy of The Correspondence Method

	N	P	R
Pattern 1	7	100	71.4
Pattern 2	7	71.4	71.4

5. Conclusion

We have proposed a method to link moving object's trajectories on multiple cameras based on the estimated angle values of the trajectories' direction. Experiment results on two cameras with two kinds of camera's placements show that the accuracy of the correspondence method is above 70%.

References

[1] O. Javed, et al., ``Tracking across multiple camera with disjoint views," The Ninth IEEE Int. Conf. on Computer Vision, Vol. 2, pp. 952-957, 2003.

[2] Y. Ikegame, M. Hirano, T. Tamaki, M. Yamamoto, ``Probabilistic estimation of pedestrian routes over non-overlapping views," Proc. of PRMU2004, pp. 186, Feb. 2005 (in Japanese).

[3] N. Ukita, T. Matsuyama, `` Real-time Cooperative Multi-target Tracking by Communicating Active Vision Agents," IPSJ SIG Notes. CVIM 2002(2), pp. 29-36, January 2002 (in Japanese).

[4] S. Noda, A. Shimada, D. Arita, R. Taniguchi, ``Tracking across non-Overlapping Cameras based on Estimated Camera Networks," Proc. of IPSJ Symp. 2007, pp. 845-850, 2007 (in Japanese).

[5] S. Weng, et al., ``Real-time Moving Object Tracking in H.264/AVC Video based on Variable Size of Macroblock and Forward Motion Vector Information," Proc. of Forum on Information Technology (FIT) 2010, pp. 413-420, 2010.