Vehicle tracking by detection in UAV aerial video

TABLE OF CONTENTS

1

Introduction

2

Tracking by detection

3

Experiments

1. Introduction

Motivation

UAV video is becoming an effective supplement to fixed monitoring video.

It has obvious advantages in the aspect of traffic information acquisition:

- Unique flexibility and easy maneuverability
- More comprehensive and clear video data
- Fast deployment in emergent situation

The purpose of this paper is to give a method of vehicles tracking in unmanned aerial vehicle (UAV) video to support further traffic information analysis.

Figure 1. some UAVs

Figure 2. some UAV images

1. Introduction

Characteristics of traffic video shooting by UVA:

Large scene, different background, containing dozens of vehicles with different size and wide-range of speed

At present, there are mainly three kinds of vehicle detection algorithms:

- Detection based on prior knowledge
- Detection based on motion information
- Detection based on machine learning
    - Not depend on shifting scene
    - Suitable for big traffic
    - High accuracy

Tracking: determine the position of the object in every frame continuously

- Tracking based on Kalman filtering
    - Small storage
    - Small computational cost
- Tracking based on Mean Shift
- Tracking based on particle filter
- on-line boosting tracking method

TABLE OF CONTENTS

1 Introduction

2 Tracking by detection

3 Experiments

2. Tracking by detection

As a tracking by detection method, we firstly detect vehicles in an UAV Video.
Here we use Faster-RCNN in following steps:

- Extract feature maps from each UAV aerial image using sharable convolutional layers.
- Generate region proposals on feature maps using RPN module.
- Train the RPN module and detector module in Fast R-CNN with shared features.

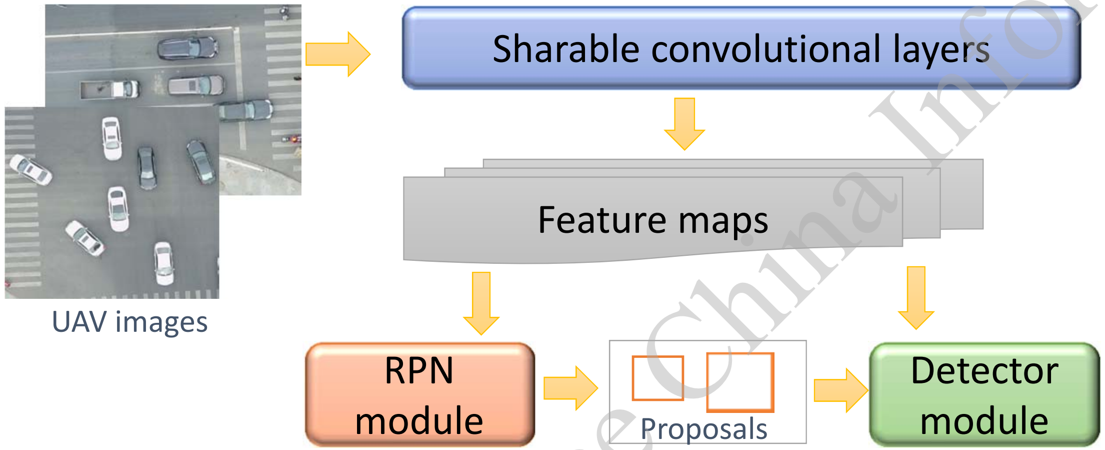

Figure 3. Working flow of Faster R-CNN

Figure 4. Mark the samples with"LabelImg"

2. Tracking by detection

To utilize Kalman filtering as a tracking method, we construct a Kalman filter motion model including equations of state and observation. During the tracking process, the time between the adjacent images is very short. We assume that the motion of vehicle in unit time is an uniform motion, the system state and the observed value are linear. So the equation of state is :

$$S(t) = A(\Delta t)S(t-\Delta t) + \omega(t-\Delta t)$$

$S(t)$ represents the state of the system at time t

$A(\Delta t)$ expresses the state transform matrix within Δt

$\omega(t)$ indicates the estimation error

We use four dimensional vectors, containing the vehicle's position and velocity, to represent the system state and the estimation error.

$$S(t)=[x(t), y(t), \vec{v}_x(t), \vec{v}_y(t)]^T$$

$$\omega(t) = [\omega_x(t), \omega_y(t), \omega_{v_x}(t), \omega_{v_y}(t)]^T$$

$$A(\Delta t) = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

In the UAV images, only the position can be observed for the state vector. We assume that the observed value are linear, so the equation of observation is:

$$O(t) = H(t)S(t) + e(t)$$

$H(t)$ is the observation matrix,

represents as:

$$H(t) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$O(t)$ is the observation vector

$e(t)$ is the observation error.

2. Tracking by detection

Tracking Algorithm

After constructing the motion model with equations of state and observation, we could track vehicles based on detection results by Kalman filtering algorithm.

First, initialize Kalman filter.
Then, run the loop of predicting the state, matching target and updating Kalman parameters.

Under the help of prediction function, Kalman filter gets trajectory for every target very fast.

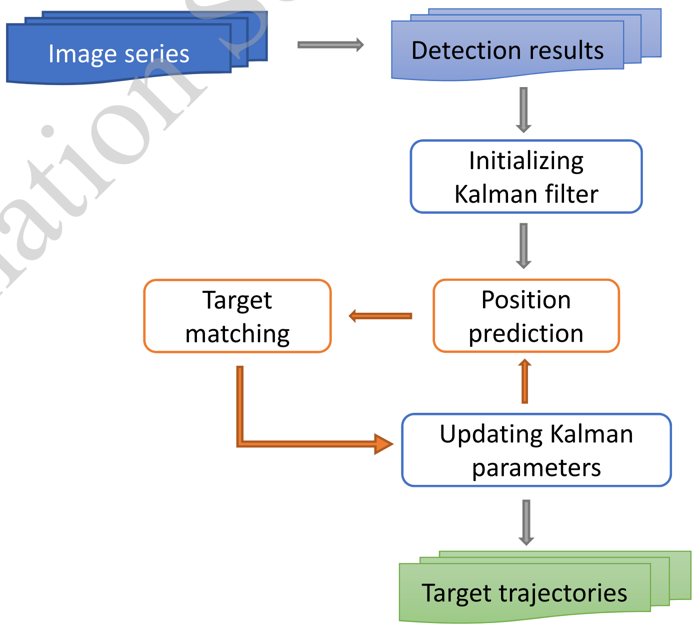

Figure 6. Processing flow of vehicle tracking by Kalman filtering algorithm

TABLE OF CONTENTS

1 Introduction

2 Tracking by detection

3 Experiments

3. Experiments

Dajiang
“Mavic”

Shoot device and scenarios:

We used a Dajiang “Mavic” UAV to get videos for our experiments.

- Shot from different fly altitudes: 76m, 86m and 96 m
- Varied traffic scenarios: crossroad, a three-way intersection, a roundabout and an exit of a main stem

Figure 7. Images from a same crossroad in different seasons with different fly altitudes
(from left to right: 76m , 86m and 96m)

3. Experiments

Shoot device and scenarios:

Dajiang
“Mavic”

We used a Dajiang “Mavic” UAV to get videos for our experiments.

- Shot from different fly altitudes: 76m, 86m and 96 m
- Varied traffic scenarios: crossroad, a three-way intersection, a roundabout and an exit of a main stem

Figure 8.   Images from different places
(from let to right: a roundabout, a three-way intersection, an exit of a main stem)

3. Experiments

Training samples:

★ We use one of the video shot from 76 meters height on a crossroad at winter as our training examples.

★ Totally: 1,314 UAV images

Benchmarks:

★ We labeled the rest videos as our benchmarks (Table 1).

★ Totally: 8 videos, totally over 11 minutes.

Evaluation criterion:

We use the recall (Re) and the precision (Pr) as the evaluation criterions of the detection results.

$$Re=TP/ (TP+FN)$$

$$Pr=TP/ (TP+FP)$$

TP is the correctly detected number of vehicles;

FP is the number of errors being mistaken as the vehicles;

FN is the number of vehicles detected as the background

Table 1. List of our benchmark Data set

	Name	Fly Height	Scene	Season
1	CR 76 Winter 2	76	A crossroad as same as the one in our training examples	Winter
2	CR 76 Winter 3	76		
3	CR 76 Summer	76		Summer
4	CR 86 Summer	86		
5	CR 96 Summer	96		
6	RA 76	76	A roundabout	
7	3WI 76	76	A three-way intersection	
8	Exit 76	76	An exit of a main stem	

3. Experiments

Results of detection:

Video Name	Faster RCNN		YOLOv3	
	Pr	Re	Pr	Re
CR 86 Summer	0.9261	0.9664	0.816	0.7901
CR 96 Summer	0.9806	0.9641	0.854	0.8071
CR 76 Summer	0.9694	0.9601	0.9243	0.9338
RA 76	0.9245	0.9807	0.6106	0.6496
3WI 76	0.908	0.964	0.8542	0.803
CR 76 Winter 2	0.9762	0.9586	0.9049	0.9013
CR 76 Winter 3	0.919	0.8891	0.8543	0.8752
Exit 76	0.9352	0.9534	0.9019	0.8774
average	0.942375	0.95455	0.840025	0.829688

3. Experiments

Results of detection:

Figure 9. Detection results for *CR 76 Winter 2* and *CR 76 Winter 3.*

3. Experiments

Results of detection:

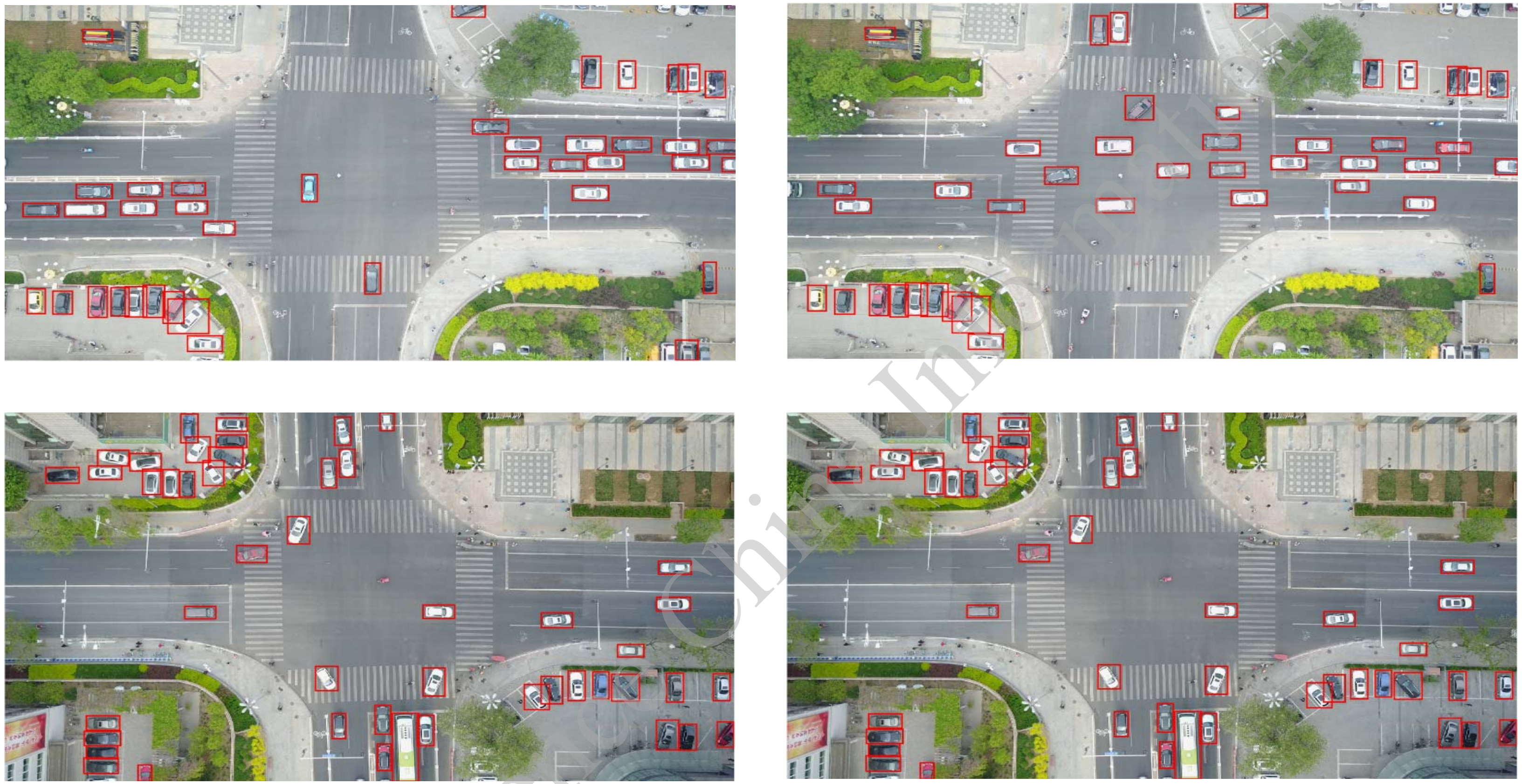

Figure 10. Detection results for *CR 86 Summer* (up) and *CR 96 Summer* (bottom).

3. Experiments

Results of detection:

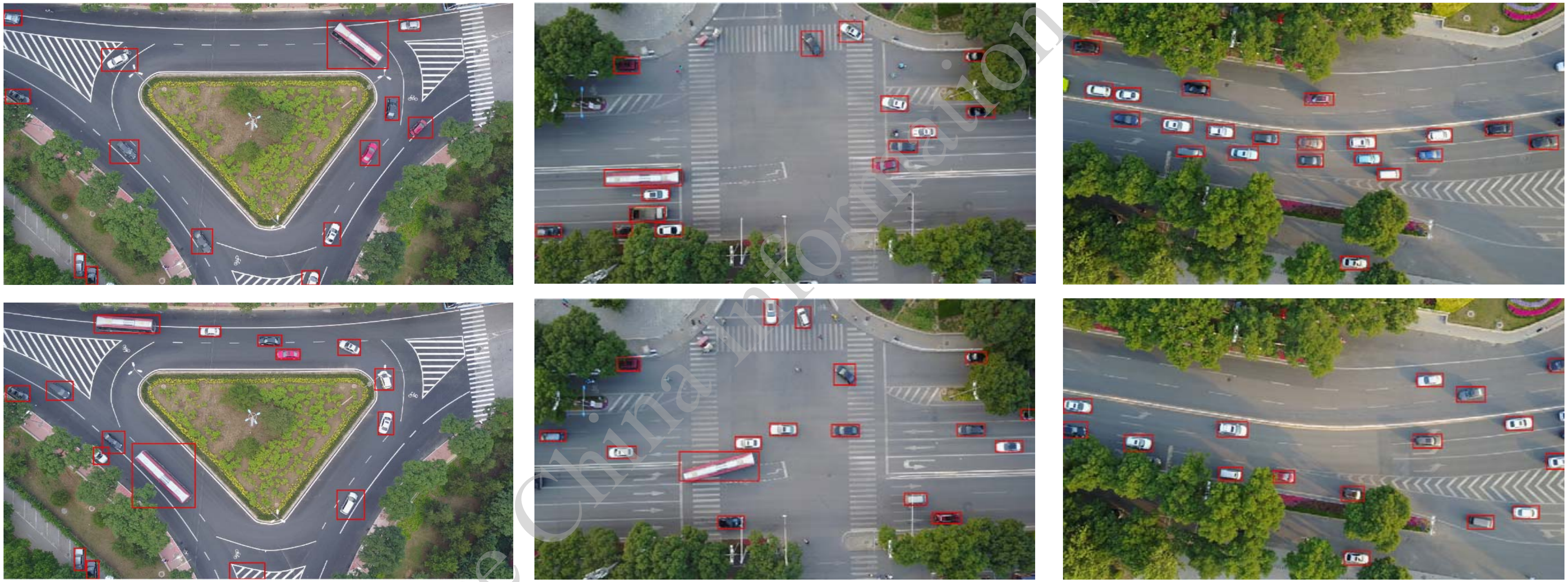

Figure 11. Detection results for *RA 76* (left), *3WI 76* (middle) and *Exit 76* (right).

3. Experiments

Results of tracking:

Video Name		CR 86 Summer	CR 96 Summer	CR 76 Summer	RA 76	3WI 76	CR 76 Winter 2	CR 76 Winter 3	Exit 76
GT		89	110	183	43	96	134	159	119
Tracking result based on Faster RCNN	MT	87	107	179	42	93	131	151	115
	PT	2	1	4	1	3	2	5	3
	ML	0	2	0	0	0	1	3	1
	IDs	58	12	138	5	37	13	96	103
	MOTA	90.47%	95.98%	92.66%	95.73%	91.11%	94.92%	81.96%	94.41%
	MOTP	0.011	0.011	0.009	0.01	0.01	0.01	0.014	0.011
Tracking result based on YOLOv3	MT	77	91	174	33	79	124	150	92
	PT	6	12	7	6	13	5	6	24
	ML	6	7	2	4	4	5	3	3
	IDs	63	33	25	41	12	72	70	91
	MOTA	61.31%	67.08%	86.39%	24.68%	69.35%	80.14%	72.70%	76.22%
	MOTP	0.172	0.167	0.163	0.145	0.189	0.152	0.163	0.177

3. Experiments

Results of tracking:

Figure 12. Tracking results of the 1st, 100th,200th, 300th frame of one video

Thank you