EasiSee: Real-Time Vehicle Classification and Counting via Low-Cost Collaborative Sensing

Rui Wang, Lei Zhang, Kejiang Xiao, Rongli Sun, and Li Cui

IEEE Transactions on Intelligent Transportation Systems, Vol. 15, No. 1, pp.414-424,

February 2014

Presented by: Yang Yu {yuyang@islab.ulsan.ac.kr}

Nov. 8, 2014

Motivation and Contribution

- This paper provides a real-time vehicle classification and counting system based on WSNs, namely, EasiSee.
  - Accurate vehicle classification.
  - Low-delay real-time performance.
  - Low resource consumption.
- Propose an event trigger mechanism-CSM(collaborative sensing mechanism), which activates the camera sensor node only when a vehicle detected, to avoid keeping the camera sensor node working all the time.
- Propose a robust vehicle image processing algorithm with low computational complexity, including the vehicle image segmentation and physical feature extraction.

Overview of EasiSee System

- Vehicle detection unit -the magnetic sensor node,
- Vehicle classification unit -the camera sensor node,
- Collaboration unit-wireless sensor network.

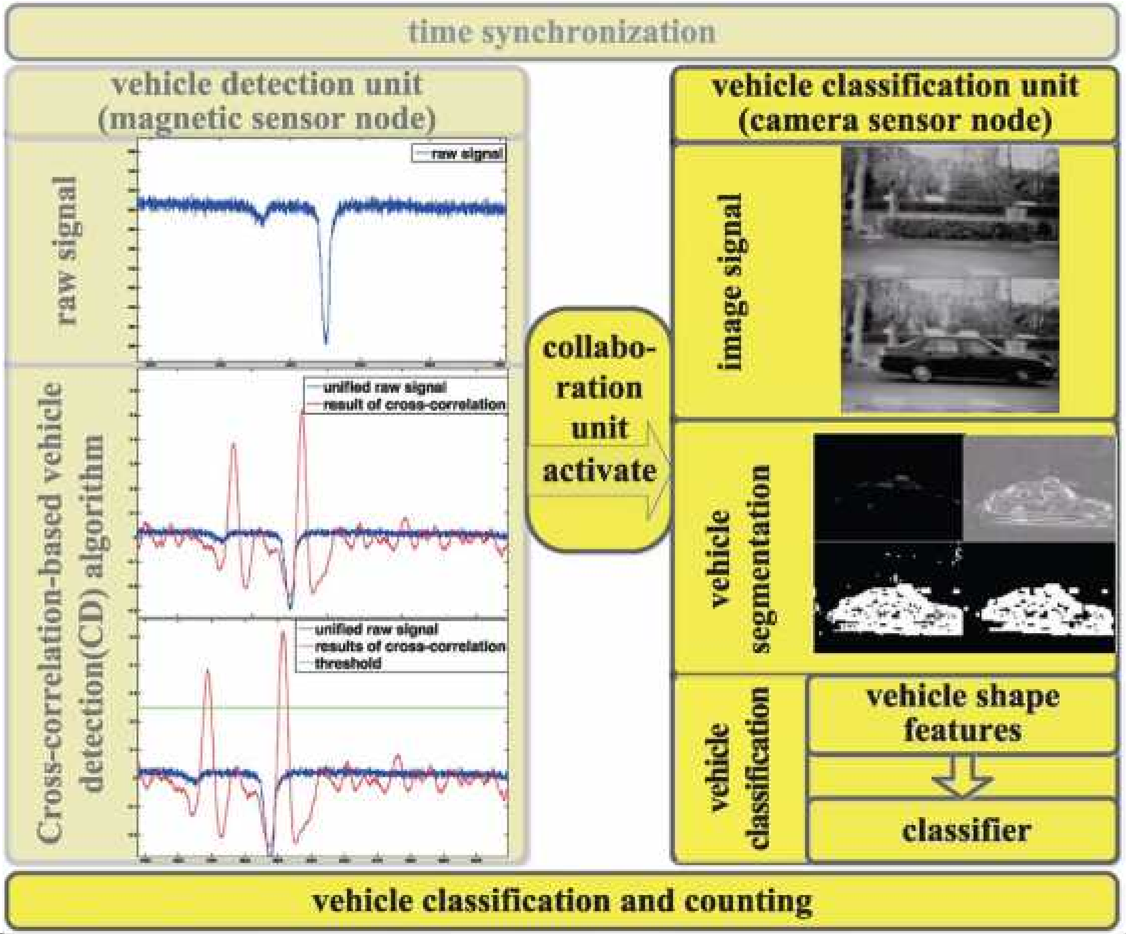
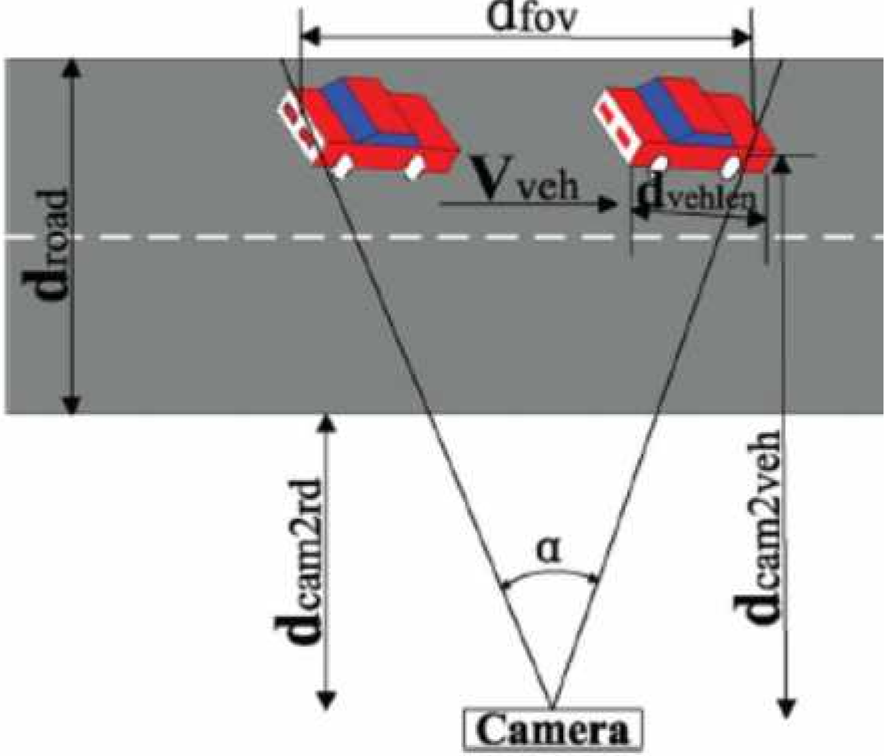


Image Capture Frequency

- The boundary condition is as follows:
  - Just enter in the first image.
  - Just leave in the second image.
- At least capture one complete car image when the speed is maximum.



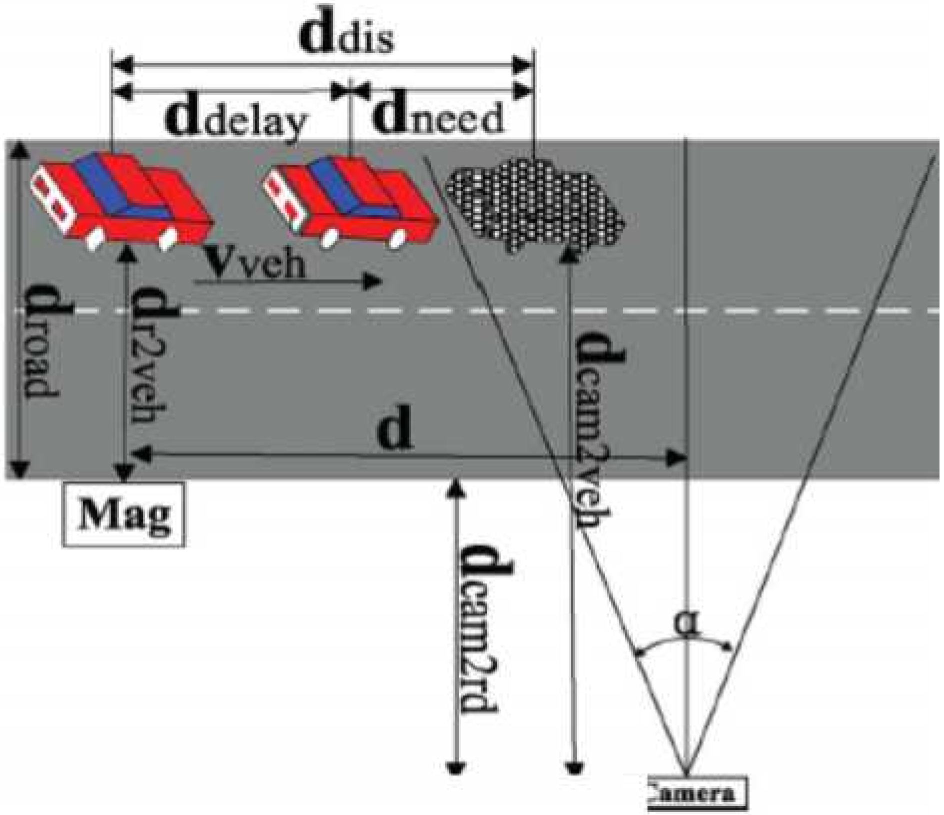
$$\begin{cases} (f_c)_{\min} = 1/(t_c)_{\max} \\ (f_c)_{\max} = 1/(t_c)_{\min} \end{cases}$$

$$\begin{cases} t_c = (d_{\text{fov}} - d_{\text{vehlen}})/v_{\text{veh}} \\ d_{\text{fov}} = 2 \times d_{\text{cam2veh}} \times \tan(\alpha/2). \end{cases}$$

$$\begin{cases} (t_c)_{\min} = \frac{2\times(d_{\text{cam2veh}})_{\min}\times\tan(\alpha/2)-(d_{\text{vehlen}})_{\max}}{(v_{\text{veh}})_{\max}} \\ (t_c)_{\max} = \frac{2\times(d_{\text{cam2veh}})_{\max}\times\tan(\alpha/2)-(d_{\text{vehlen}})_{\min}}{(v_{\text{veh}})_{\min}} \end{cases}.$$

Number of Image to Capture

- The boundary condition is as follows:
  - Not fully covered after the camera is activated.
- The n th image can capture car image when the speed is minimum.



t_{img} is equal to $(t_c)_{\min}$

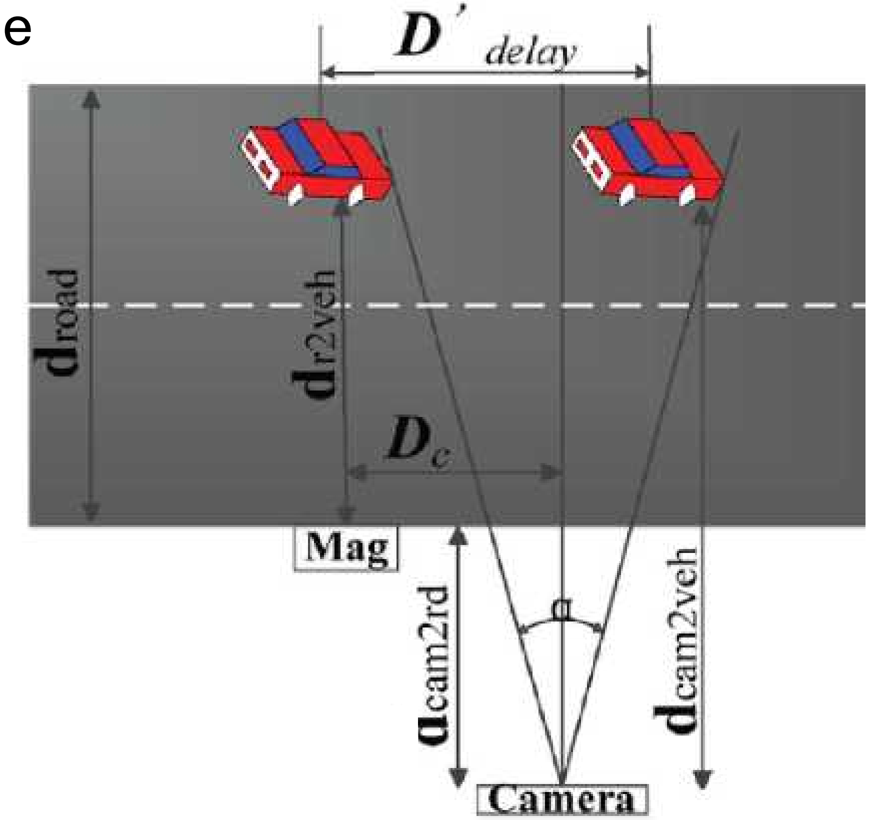
$$N_{\text{need}} = t_{\text{need}}/t_{\text{img}}$$

$$\begin{cases} t_{\text{need}} = (d_{\text{dis}} - d_{\text{delay}})/v_{\text{veh}} \\ d_{\text{dis}} = d - d_{\text{fov}}/2 + d_{\text{vehlen}}/2. \end{cases}$$

$$\begin{aligned} n &= \max(N_{\text{need}}) \\ &= (d - \min(d_{\text{cam2veh}}) \times \tan(\alpha/2) + \max(d_{\text{vehlen}})/2) \\ &\quad / (t_{\text{img}} \times \min(v_{\text{veh}})) - t_{\text{delay}}/t_{\text{img}}. \end{aligned}$$

Distance between the Sensors

- The boundary condition is as follows:
  - The front end just leaving when the camera captures the first image.
- This distance make the first image can capture car image when the speed is maximum.



$$d'_{\text{delay}} = t'_{\text{delay}} \times v_{\text{veh}}$$

$$D_c = d'_{\text{delay}} - d_{\text{fov}}/2 + d_{\text{vehlen}}/2$$

$$\begin{aligned} d_{\text{cam2mag}} &= \max(D_c) \\ &= t'_{\text{delay}} \times \max(v_{\text{veh}}) + \max(d_{\text{vehlen}})/2 \\ &\quad - \min(d_{\text{cam2veh}}) \times \tan(\alpha/2) \end{aligned}$$

Mean Compression

- Propose a low-complexity but robust vehicle segmentation algorithm to solve the illumination variation and low gray difference problems.
- The mean compression reduces most of the random scatter noise.

$$\mathrm{cImg}(i,\ j) = \sum_{s=s_l}^{s=s_h} \sum_{t=t_l}^{t=t_h} \mathrm{rImg}(s,\ t) \Big/ \mathrm{bRow} \times \mathrm{bCol}$$

where $i = 0, 1, \ldots, (\mathrm{cRow} - 1)$, $j = 0, 1, \ldots, (\mathrm{cCol} - 1)$, cImg is the new compressed image with size $\mathrm{cRow} \times \mathrm{cCol}$, the size of every small piece is $\mathrm{bRow} \times \mathrm{bCol}$, rImg is the image directly from the camera, $s_l = \mathrm{bRow} \times i$, $s_h = \mathrm{bRow} \times i + \mathrm{bCol}$, $t_l = \mathrm{bCol} \times j$, and $t_h = \mathrm{bCol} \times j + \mathrm{bRow}$.

- The image size can be reduced by bRow $\times$ bCol.

Difference Image

- Difference image by background subtraction:

$$\text{subImg}(i, j) = abs\left(v\text{Img}(i, j) - \text{bImg}(i, j)\right)$$

where $i = 0, 1, \ldots, (\text{row} - 1)$, and $j = 0, 1, \ldots, (\text{col} - 1)$.

subImg is the difference image, vImg is the vehicle image, and bImg is the background image. The size of the image is row $\times$ col.

Gradient Image

- Gradient Image by the First-Order Gradient Operation to the Difference Image:

$$\mathrm{dxSImg}(i, j) = \mathrm{subImg}(i, j+1) - \mathrm{subImg}(i, j)$$

$$\mathrm{dySImg}(i, j) = \mathrm{subImg}(i+1, j) - \mathrm{subImg}(i, j)$$

$$\mathrm{dfSImg}(i, j) = |\mathrm{dxSImg}(i, j)| + |\mathrm{dySImg}(i, j)|$$

dxSImg is the first-order horizontal gradient image, dySImg is the first-order vertical gradient image, and dfSImg is the first-order gradient image.

- Replace the square and extraction operations in the gradient computation by the straight add operation to the absolute values.

Segmentation by Threshold

- Vehicle Segmentation by Thresholds:

$$\text{bwDSImg}(i, j) = \begin{cases} 1, & \text{if dfSImg}(i, j) \geq \text{Th} \\ 0, & \text{if dfSImg}(i, j) < \text{Th} \end{cases}$$

where $\text{Th} = \mu + 3\sigma$, μ is the expectation of the gradient image, and σ is the approximated value of the standard deviation.

Extraction of Shape Features

- The length of the longest nonzero virtual row(column) serial will be the physical length of that vehicle object.

$$\begin{cases} \mathrm{virtualRow}(j) = \sum_{i=0}^{\mathrm{row}-2} \mathrm{bwDSImg}(i, j) \\ \mathrm{virtualCol}(i) = \sum_{j=0}^{\mathrm{col}-2} \mathrm{bwDSImg}(i, j) \end{cases} \quad \begin{cases} L_{\mathrm{veh}} = \max\{\mathrm{length}(\mathrm{RL}_i)\} \\ H_{\mathrm{veh}} = \max\{\mathrm{length}(\mathrm{RH}_j)\} \end{cases}$$

- The first “1” and last “1” of each row(column) use to outline the vehicle profile. We can calculate the vehicle perimeter.

P_{veh}

Vehicle Classification

$$LHR = L_{veh} / H_{veh}$$

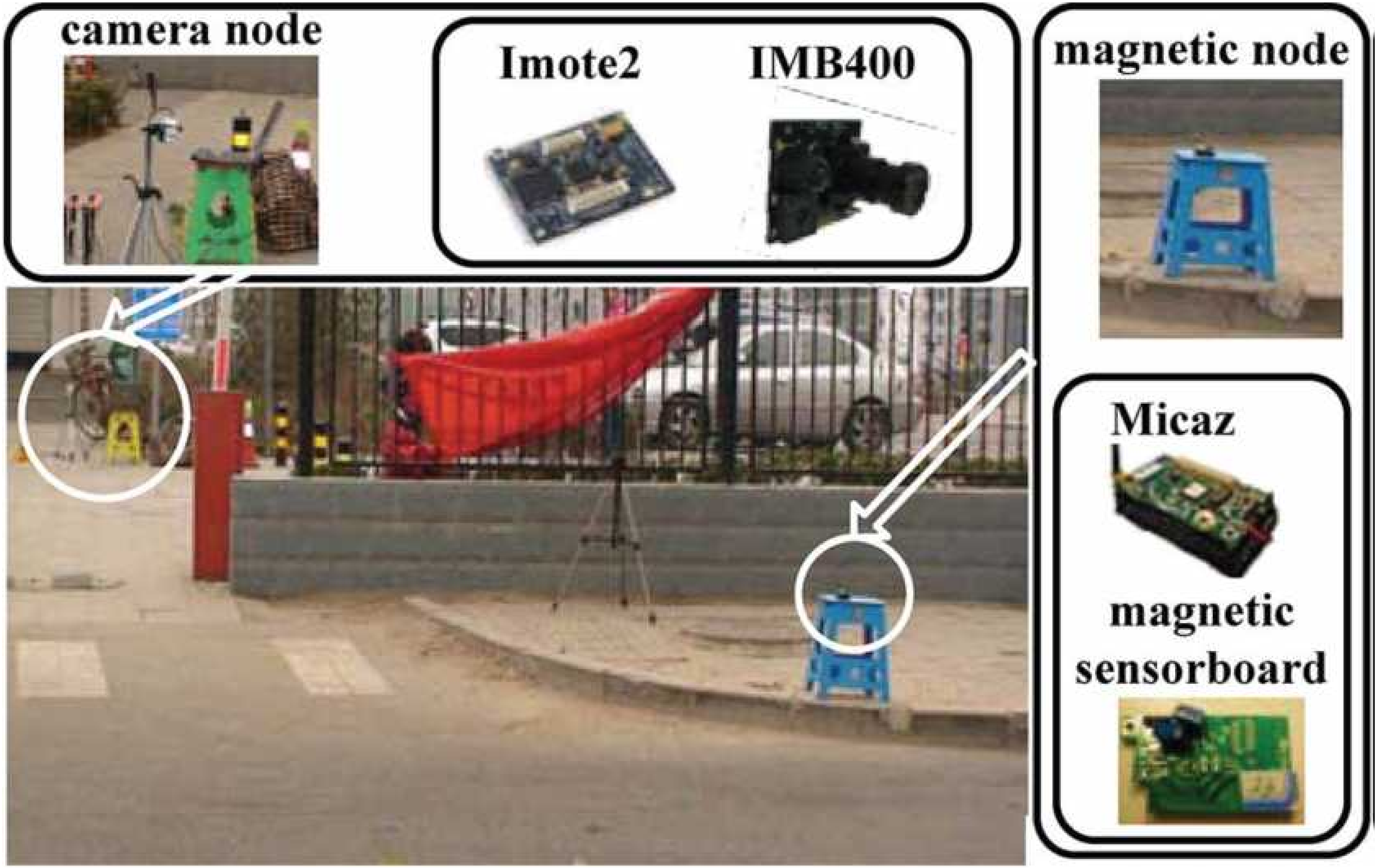
$$PSR = P_{veh} \times P_{veh} / L_{veh} / H_{veh}$$

- Based on the training set, the LHR threshold is set to be 1.5. We use it to classify the bicycles and the motor.

- Based on the training set, the center of two classes are $\overline{X}^{(\mu_1)}$=(1.8, 8.68), and $\overline{X}^{(\mu_2)}$ = (1.62, 2.68). The i th sample is $X_i = (LHR_i, PSR_i)$ We use Euler distance (ED) to classify the car and the minibus.

Experimental Environment

- Deployment scene on real road environment:



Typical Vehicle Scenes

- Typical vehicle scenes. (a) Minibus. (b) Family car.
  (c) Family car with low gray difference with its background.
  (d) Bicycle with low gray difference with its background.

(a) (b) (c) (d)

Experimental Parameters

- Based on the statistics on the captured traffic data, we assign values for the following parameters:

  ~ is $25°$, t'_{delay} is 0.6 s, d_{vehlen} is 4–5 m, d_{delay} is 3.6–4 m, d_{vehhei} is 1.5–1.8 m, $^{3}{}_{\text{veh}}$ is 5.6–11 m/s, d_{cam2veh} is 13.5–15.5 m.

  We calculate f_c is 2–11.8 Hz, set to 10 Hz.

  n is 4, set to 10.

  d_{cam2mag} is 6.18 m, set to 7 m.

Width of road d_{road}	7m
Sensing range of magnetic sensor node d_{msen}	7m
Frequency of magnetic sensor f_m	100Hz
The distance from the camera sensor node to the near roadside $d_{cam2road}$	10m
The height of the camera sensor node h_c	0.6m
Frequency of Sony Handycam HDR-SR5 f_s	24 frame/s

Vehicle Detection Accuracy

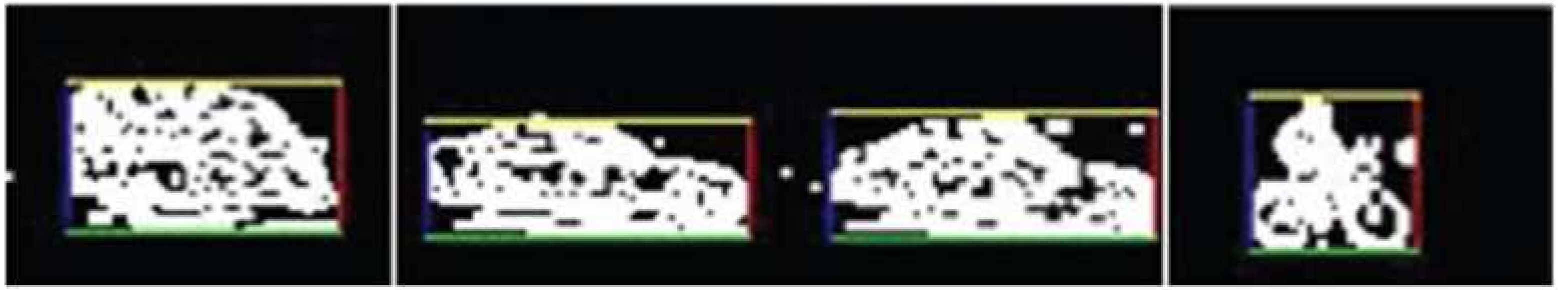
- Vehicle detection accuracy of magnetic sensor node.

VEHICLE DETECTION RESULTS

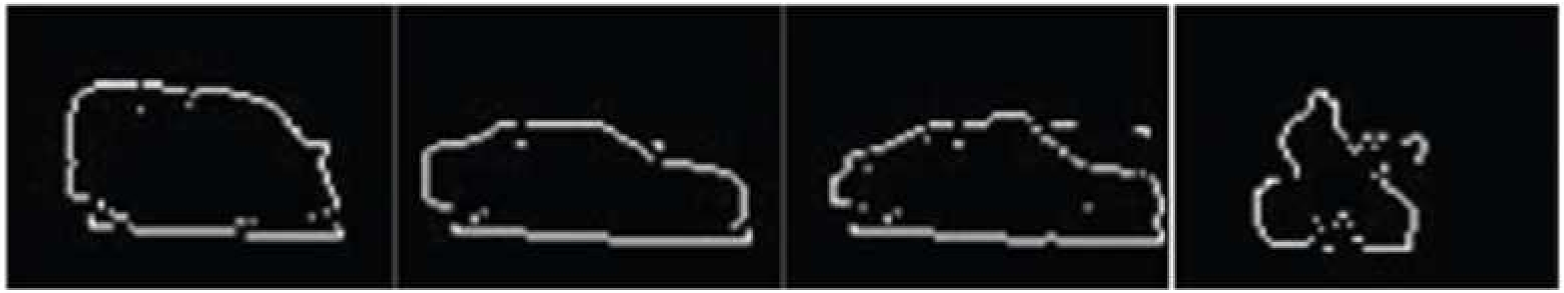
	Bicycle		Car		minibus	
	HDR-SR5	Detect $(r/n*)$	HDR-SR5	Detect $(r/n*)$	HDR-SR5	Detect $(r/n*)$
Group1	24	15/9	56	52/4	14	13/1
Group2	12	8/4	62	58/4	9	9/0
Group3	15	10/5	81	76/5	12	12/0
Total	51	33/18	199	186/13	35	34/1
detection accuracy	64.71%		93.47%		97.14%	
			95.31%			

Segmentation Algorithm Image

- Bounding box and vehicle outline.
- The image of 80 $*$ 60 pixels can still provide an ideal vehicle segmentation result.



(a)



Accuracy of Vehicle Classification

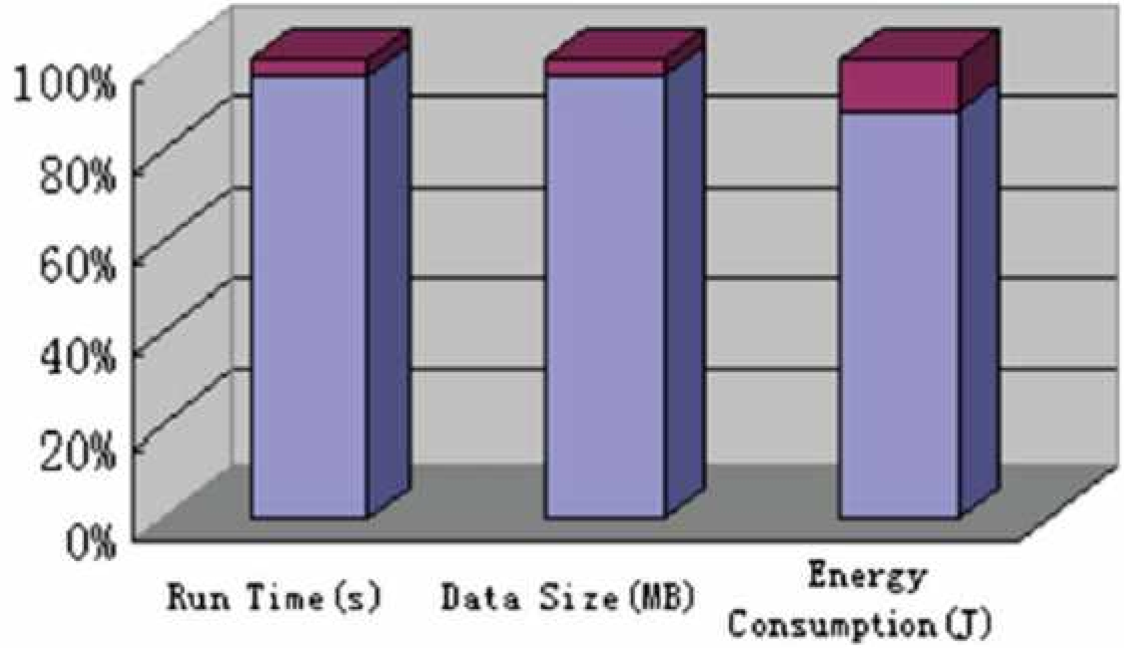
- Accuracy of vehicle classification :

CLASSIFICATION RESULTS

Category	Sample number	Classification		Classification accuracy
		Number	Real type (b/c/m^*)	
bicycle	259	256	256/0/0	98.84%
car	140	141	2/134/5	95.71%
minibus	32	35	1/7/27	84.38%

Note:b/c/m in the table represents bicycle/car/minibus

- Analyze and compare resource consumption between Easi See and the noncollaborative method (NCM).



	Run Time (s)	Data Size (MB)	Energy Consumption (J)
EasiSee	431	4669.48	455.24
NCM	12133	131384.44	3603.5

Conclusions

- EasiSee reduces the energy consumption while providing comparable classification accuracy with respect to NCM systems.
- As all computations are carried out at local node, EasiSee is proved to be a distributed system with good scalability.
- Future work
  - how to process the image in which multiple targets are captured.
  - how to improve this system to be practical on a large scale.