SakuraSensor: Quasi-Realtime Cherry-Lined Roads Detection through Participatory Video Sensing by Cars

Shigeya Morishita†, Shogo Maenaka†, Daichi Nagata†
Morihiko Tamai†, Keiichi Yasumoto†, Toshinobu Fukukura‡, Keita Sato‡
†Nara Institute of Science and Technology
Ikoma, Nara 630-0192, Japan
{morishita.shigeya.mk6, nagata.daichi.mr3, maenaka.shogo.me4, morihi-t, yasumoto}@is.naist.jp
‡DENSO CORPORATION
Kariya, Aichi 448-8661, Japan
{TOSHINOBU_FUKUKURA, KEITA_SATO}@denso.co.jp

ABSTRACT
In this paper, we propose SakuraSensor, a participatory sensing system which automatically extracts scenic routes information from videos recorded by car-mounted smart-phones and shares the information among users in quasi-realtime. As scenic routes information, we target flowering cherries along roads since the best period of flowering cherries is rather short and uncertain from year to year and from place to place. To realize SakuraSensor, we face two technical challenges: (1) how to accurately detect flowering cherries and its degree, and (2) how to efficiently find good places of flowering cherries (PoIs) using the participatory sensing technique. For the first challenge, we develop an image analysis method for detecting image pixels that belong to flowering cherries. To exclude artificial objects with similar color to flowering cherries, we also employ fractal dimension analysis to filter out unnecessary image areas. For the second challenge, we propose a method called *k-stage sensing*. In this method, the interval for sensing (taking a still image and applying the image analysis) by each car is dynamically shortened so that the roads near the already found PoIs are more densely sensed. We implemented SakuraSensor consisting of client-side software for iOS devices and server-side software for a cloud server and conducted experiments to travel cherry-lined roads and record videos by several cars. As a result, we confirmed that our method can identify flowering cherries at about 74 % precision and 84 % recall. We also confirmed that our k-stage sensing method could achieve the comparable PoI detection rate with half sensing times compared to a conventional method.

Author Keywords
Participatory sensing; flowering cherries detection; image analysis; k-stage sensing.

ACM Classification Keywords
I.4.9 Image Processing and Computer Vision: Applications

INTRODUCTION
Car navigation and routes recommendation services play an important role for providing drivers with comfortable and efficient driving. In addition to ordinary services that only consider traveling time or fuel efficiency of routes, scenic routes recommendation services have begun to be provided, such as scenicbyways.info [1], NAVITIME [2] or Honda's internavi [3] in recent years.

In scenic routes recommendation services, the user should be able to get enough information about the recommended routes so that he/she can easily understand what kind of (and how beautiful) scenery will be able to view when traveling these routes. Regarding this point, in the existing services, only the information including texts and images are provided for the users. Moreover these information are available only for the pre-determined spots among the limited number of scenic routes. These limitations are caused by the fact that the information are edited manually by the service provider, and the cost, especially man-power resource will be significantly increased to improve the service quality. For this reason, in the existing services, the number of pre-determined spots is relatively small and information update frequency on the spots tends to be low. However, since the view of the scenery along a route possibly varies depending on time, weather, season, traveling direction and so on, it is desirable that the scenery information is updated frequently. Moreover, the scenery information that consist of texts and images are not sufficient for the users to assess how good each route is, because it is difficult to intuitively grasp what kind of scenery will be able to view in advance from out-of-date still images or texts.

To address these problems, we can leverage a participatory sensing technique [4] to collect information about scenic routes from users' car-mounted smartphones and share the collected information among the users. Here, each smartphone mounted at a car continuously records the view from the windshield as a sequence of videos. To detect good scenery, image analysis for the recorded videos is performed locally in the smartphone to quantify *scenery goodness*, i.e., how beautiful scenery is recorded in the videos. Then, the quantified scenery goodness is sent to the server in the cloud. The server finds locations with especially good scenery by comparing among quantified values of scenery goodness of various routes received from multiple traveling cars, and collects short videos recorded at these locations.

The aforementioned system allows the service provider to automatically collect information of scenic routes with a wide variety of conditions (different time period, weather, etc.) over widespread locations. Based on the collected information, a scenic routes recommendation service can be provided, where the users search the best scenic route to a specified destination and view short videos along the searched route. By this service, the user can understand the goodness of each scenic route more intuitively compared to the services that provide information based only on texts and images.

Basically, scenic beauty perceived by humans while driving depends on many factors including geographic features (e.g., landscape along a coast or a river, high altitude places) and existence of specific objects (e.g., plants, flowers, famous natural objects, famous artificial structures). Among various kinds of scenery along roads, in this paper, we target cherry-lined roads to be detected, since it seems relatively easy to achieve the quantification of scenery goodness by color analysis. Moreover, since the best period of flowering cherries is short (i.e., less than two weeks) and uncertain from year to year and from place to place, we believe that detecting cherry-lined roads in quasi-realtime is worth tackling.

In this paper, we propose *SakuraSensor*[1], a participatory sensing system which automatically identifies cherry-lined road segments from videos recorded by car-mounted smartphones and shares the information as well as short videos among users (e.g., car drivers) in quasi-realtime. To realize SakuraSensor, we face two technical challenges: (1) how to accurately detect flowering cherries and its degree, and (2) (2) how to efficiently find good places of flowering cherries (PoIs) using the participatory sensing technique.

For the first challenge, we employ two simple image analysis techniques: histogram-based color analysis and region-based fractal dimension analysis. For the color analysis, a color histogram that represents the color distribution of flowering cherries is generated in advance using various images that contain flowering cherries. Then, multiple frames are sampled from each video recorded by a car-mounted smartphone, and for each pixel of each frame, the frequency that the pixel belongs to flowering cherries is calculated. Next, the numerical value that quantifies how much flowering cherries are contained in a frame, which is called *cherry intensity* hereafter, is calculated as the mean value of the frequency in the frame. It should be noted that flowering cherries look very different depending on the lighting conditions (direct light, backlight, and intermediate). To cope with this problem, we employed a technique to use only a part of HSV color space for analysis. In addition, to avoid misdetection of objects (e.g., artificial structures) that contain the similar colors in flowering cherries, SakuraSensor divides each frame to multiple image regions and applies to each region fractal dimension analysis [5] to investigate if the region has complex edge patterns or not. Histogram-based analysis is applied to regions only with complex edge patterns which commonly appear in the regions of leaves and trees.

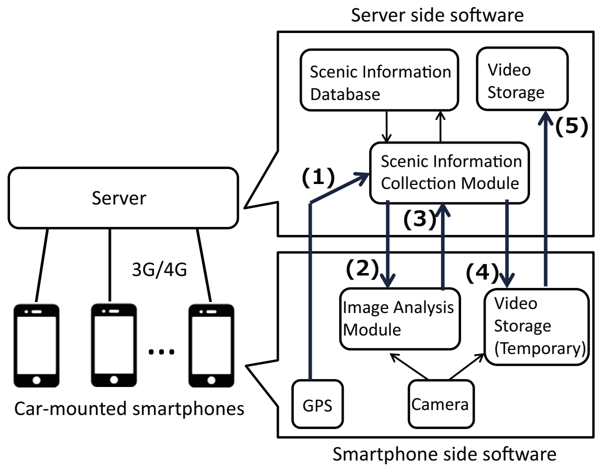


Figure 1. Overall architecture of proposed participatory sensing system

For the second challenge, aiming to reduce the resource (e.g., battery, computation power, storage, 4G/LTE communication) consumed by each participant, we propose a method called *k-stage sensing* where k is the number of stages. In the proposed method, first each smartphone mounted at a car senses (i.e., takes a still image and applies the image analysis) at regular intervals with a long distance. When flowering cherries are detected in the image, the place is registered as a PoI to a cloud server, and other cars passing near the PoIs sense at shorter distance intervals. The interval is narrowed step by step until reaching k-th stage. Finally, the cloud server identifies particularly good PoIs over a predefined threshold and ask cars passing near those PoIs to record and upload videos so that the videos are shared among users.

We implemented SakuraSensor consisting of client-side software for iOS devices and server-side software for a cloud server. The client software implements the proposed image analysis method by using OpenCV. We conducted experiments to travel cherry-lined roads and record videos by several cars. Using the collected videos, we evaluated the accuracy of identifying flowering cherries by our image analysis method. As a result, we confirmed that our method can identify flowering cherries at about 74 % precision and 84 % recall. We also evaluated our k-stage sensing method and confirmed that our method achieves the PoI detection rate comparable to the fixed interval sensing method with about half times of sensing.

SAKURASENSOR: OVERALL ARCHITECTURE

The proposed participatory sensing system collects scenic routes information from a number of car-mounted smartphones while they are traveling. The information to be col-

[1]Sakura means cherry in Japanese.

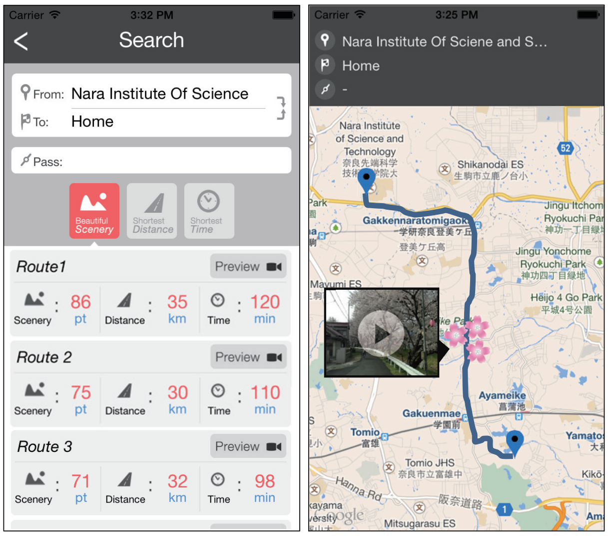
lected consist of GPS logs, cherry intensity, and short videos recorded at locations with high cherry intensity. These information are uploaded to a server and utilized to provide services such as scenic routes recommendation. Fig. 1 shows the overall architecture of the system, which consists of the application running on a smartphone and the software running on the server.

At the smartphone side, while a car is traveling, the smartphone continuously records GPS logs and videos captured by its camera. The GPS logs are transferred to the server without any special treatments. On the other hand, for the recorded videos, it would be impossible to continuously transfer them to the server due to the limitation on the communication bandwidth of 3G/4G network. Even if it is possible, it still imposes a significant load on the network and will not be acceptable.

To cope with this limitation, in the proposed system, instead of uploading the whole video immediately after recording it, the smartphone side software extracts *short videos* (e.g., 10 seconds) from recorded videos and uploads them based on the requests by the server. More specifically, short videos are collected in the following steps.

- First, the smartphone side software uploads GPS data to the server (arrow (1) in Fig. 1), and receives a list of locations where sensing is required (arrow (2) in Fig. 1), which is created by the server based on k-stage sensing method explained in detail in later sections. Then, at the nearby locations in the list, the smartphone side software analyzes the video stream captured by smartphone's camera using the image analysis method proposed in the next section and at the same time the video stream is stored in the smartphone's storage (every time interval, e.g., one minute, of the video stream is stored as a file).
- It extracts frames from the video stream at a specific sampling rate and calculates cherry intensity of the frames using an image analysis method (details are given in the next section).
- Cherry intensity is continuously calculated and uploaded to the server (arrow (3) in Fig. 1).
- Next, the server side software detects locations (*PoI*: Points of Interest) that have especially high values of cherry intensity by statistically analyzing the cherry intensity collected from a number of cars, and sends requests for the short videos to the smartphones that recorded videos including the specified PoI (arrow (4) in Fig. 1). A method for appropriately updating cherry intensity (overwrite, average, discard, etc) for PoI from the information collected by multiple users is out of scope of this paper and part of future work.
- Finally, the smartphone which received a request extracts the short video including the PoI specified in the request and uploads the video to the server (arrow (5) in Fig. 1).

To mitigate storage usage, the smartphone side software limits the capacity of the usable storage for recorded videos, and videos that contain frames with relatively higher values of cherry intensity are stored preferentially within the storage



(a) Route Selection View (b) Route Information View

Figure 2. User interface for scenic routes search service

capacity. Moreover, the videos that have been completed upload to the server are removed immediately.

As an example of services that use collected information of cherry intensity, we consider a scenic routes recommendation service here. Fig. 2 is an example of the user interface of the service. The route selection view (Fig. 2 (a)) shows multiple candidates of routes between the departure point and the destination point, and these routes are sorted based on cherry intensity (scenery goodness), distance, and estimated time required to travel. In Fig. 2 (a), sorting is performed based on cherry intensity. When the user selects a route, the detailed information are shown on the screen (Fig. 2 (b)) and the user can check the locations of PoI and view short videos recorded at the PoI.

To provide such a service, it is important to be able to calculate cherry intensity with sufficiently high accuracy from video streams. The next section presents the image analysis method in details to calculate cherry intensity.

DETECTION OF FLOWERING CHERRIES

This section presents the image analysis method which takes an image (i.e., a frame sampled from a video stream) as an input and outputs its cherry intensity, which quantifies the amount of flowering cherries in the image. The proposed method mainly consists of two image analysis modules: (1) histogram-based color analysis module to detect image pixels that belong to flowering cherries and (2) region-based fractal dimension analysis module to filter out unnecessary image regions that have relatively simple edge patterns such as buildings and roads. We chose these image analysis methods, because they are relatively lightweight and appropriate for realtime image analysis on smartphones, which have limited computational and battery resources. The details of these modules are described in the following subsections, respectively.

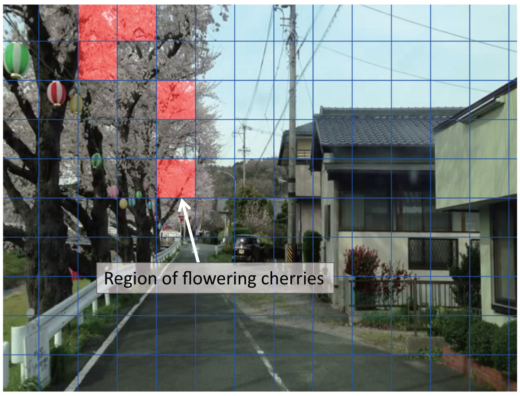


Figure 3. A tool for extracting image regions of flowering cherries

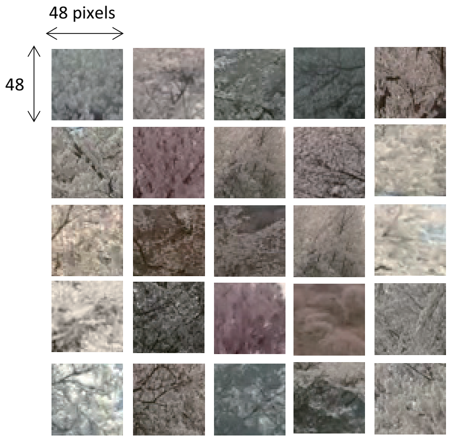


Figure 4. 25 examples of extracted cherry regions

Histogram-based color analysis for detecting flowering cherries

The color analysis module takes an image of RGB color space as an input and for each pixel contained in the input image, it quantifies to what extent the pixel has a similar color that frequently appears in the image regions of flowering cherries. To do this, we prepare a color histogram in advance, which represents frequency distribution of colors of pixels that actually belong to the regions of flowering cherries.

To create the histogram, we collect a number of videos that are captured while traveling cherry-lined roads and extract image regions that contain flowering cherries. Here, the size of each region is 48×48 pixels, for example. Extracting image regions manually from videos is a labor-intensive and time-consuming task. To support the task, we developed a tool that takes a video file as an input and saves as files specified regions of each frame contained in the video while displaying each frame. A screenshot of the tool is shown in Fig. 3. The tool draws grid lines onto each frame, and the user can select each rectangle region in the grid by clicking with the mouse. The regions selected are marked with red color. For example, in Fig. 3, five regions are currently selected. After selecting some regions, by pressing the 's' key, the selected regions are saved as separated image files. When

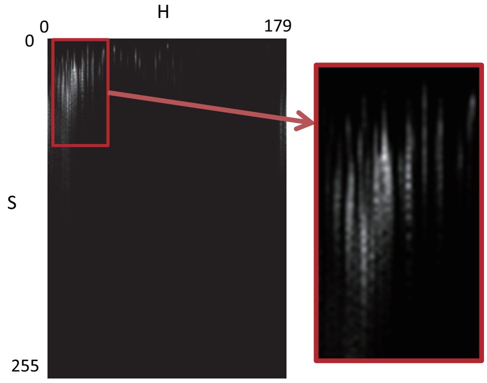


Figure 5. An example of H-S histogram generated from multiple cherry regions

using the tool, we carefully chose regions so that each region entirely contains flowering cherries and does not contain backgrounds such as sky. Hereafter, each region collected by the tool is called *cherry region*. Examples of cherry regions collected are shown in Fig. 4.

Next, we generate a color histogram that represents the frequency distribution of colors of pixels that appear in cherry regions from a number of cherry regions collected above. First, the color space of each cherry region is converted to the HSV color space. Then, for each cherry region, we build a two-dimensional color histogram (consisting of bins with some fixed value ranges) of H (Hue) and S (Saturation) components, in which the value stored in each bin is the number of pixels that are within the range of the bin. For V (Value or Brightness) component, we found that the inclusion of V into the color histogram does not contribute to or slightly degrades the accuracy of cherry intensity from preliminary experiments, because the value of V varies significantly depending on the lighting (direct or back) and weather conditions. For this reason, we do not include V into the histogram. After that, H-S histograms of all cherry regions are merged into one H-S histogram by accumulating the frequencies (values) of corresponding bins among all histograms. Lastly, the values in each bin are normalized by dividing with the maximum value among the bins. Fig. 5 shows an example of the H-S histogram built from total of 148 cherry regions. In Fig. 5, we can see that the H and S values that appear in cherry regions are biased toward the certain part within a range which is surrounded by red lines. Hereafter, when the value of H is h and S is s, the normalized value of the bin that corresponds to the vector (h, s) is denoted by $F(h, s)$.

Finally, cherry intensity of the input image is calculated using the H-S histogram obtained above. For this, we use the backprojection [6] method, which is a well-known method for object recognition in images. For each pixel $I_{i,j}$ contained in the input image, where i and j are the x and y coordinates of the pixel, respectively, we denote the values of H and S by $H(I_{i,j})$ and $S(I_{i,j})$, respectively. Then, the cherry intensity of the pixel, denoted by $D^{pix}(I_{i,j})$, is calculated by the following equation:

$$D^{pix}(I_{i,j}) = F(H(I_{i,j}), S(I_{i,j})). \tag{1}$$

Figure 6. An input image

Figure 7. A resulting grayscale image of the backprojection method

Here, the higher the cherry intensity of a pixel is, the higher the probability that the pixel actually belongs to the flowering cherries. Finally, we calculate the cherry intensity of an input image I denoted by $D^{img}(I)$ as the mean value of the cherry intensity of all pixels in I by the following equation:

$$D^{img}(I) = \frac{\sum_i \sum_j D^{pix}(I_{i,j})}{N(I)}, \tag{2}$$

where $N(I)$ is the number of pixels in I.

Fig. 7 shows the grayscale image where each pixel is represented by thickness corresponding to cherry intensity value, which is obtained by applying the backprojection method to the input image shown in Fig. 6. In this figure, we clearly see that the pixels that belong to the flowering cherries have relatively high values.

Region-based fractal dimension analysis

The histogram-based color analysis described above is not enough to calculate cherry intensity accurately, because the cherry intensity can be high incorrectly when there exist objects (e.g., structures, signboard) that contain similar colors to flowering cherries. In this subsection, we describe a fractal dimension analysis method to identify cherry regions, which have relatively complex edge patterns.

Self-similarity is the property that a similar pattern to the original pattern will appear in a part of the original pattern when magnifying. An object that exhibits self-similarity is called

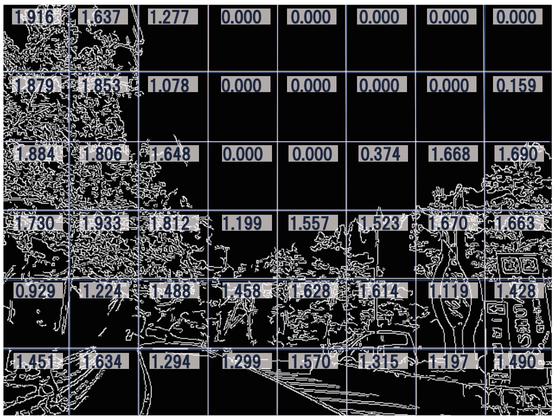


Figure 8. A result of the fractal dimension analysis

fractal, and it is known that fractals are easily found in nature such as coastlines [5]. One of the indices that characterizes fractal patterns is the fractal dimension, and it is used to quantify the complexity of the patterns in images and is applied to analyzing medical images obtained with various imaging modalities such as ultrasound and computed tomography (CT) [7].

In the proposed method, fractal dimension analysis is used as follows. First, an edge detection algorithm is applied to the input image, producing the resulting binary image. Then, the binary image is divided by a grid pattern into square blocks of the same size, and the fractal dimension is calculated for each square region. Lastly, the color analysis explained in the previous sub-section is applied to only the regions whose fractal dimension is higher than a specific threshold (called *fractal dimension threshold*). Among several methods for calculating fractal dimension, we employ the box counting method [5] because it is easy to implement.

Fig. 8 shows the fractal dimension of each square region in the input image shown in Fig. 6. In the proposed method, Canny's algorithm [8] is used for edge detection. The size of the square is empirically set to 80×80 pixels. The figure suggests that the square regions including flowering cherries have complex edge patterns and they have relatively higher values of fractal dimension than the other regions.

Computation of cherry intensity of a video

In this subsection, given a video m, we describe how to calculate the cherry intensity of m, which is denoted by $D^{mov}(m)$. First, the frames are extracted from m with a specific sampling rate. Here, the set of frames extracted is denoted by $F(m)$. Then, fractal dimension analysis is applied to each frame, and if there are some square regions whose values of fractal dimension are higher than the fractal dimension threshold, the cherry intensity of the pixels within the square regions are calculated by Equation (1), otherwise the cherry intensity of the frame is 0. Finally, $D^{mov}(m)$ is calculated as the mean value of the cherry intensity of all frames and calculated by the following equation:

$$D^{mov}(m) = \frac{\sum_{I \in F(m)} D^{img}(I)}{|F(m)|}. \quad (3)$$

The image analysis methods proposed in this section can be directly applied to the similar types of scenery like autumn leaves in fall and fresh green leaves in spring by creating a color-histogram dedicated to the specific scenery type.

IMPLEMENTATION

In this section, we describe the details of implementation of our participatory sensing system.

Overview of implementations

We have implemented the smartphone side software for iOS (version 8.0.2) and the server side software for Linux. The server side software has been implemented in JavaScript using Node.js[2]. We use WebSocket [9] and its library ws[3] for communication to the client side software. The client side software uses SocketRocket[4] which is a library for WebSocket for communication to the server side software. It also incorporates the SakuraSensor module which is implemented in Objective-C using OpenCV for iOS version 3.0.0-alpha.

Performance of short video uploads

The proposed participatory sensing system aims to achieve quasi-realtime (e.g., several minutes delay is permitted) sharing of cherry intensity and short videos collected while users are traveling by car.

We conducted experiments of uploading 10-second short videos from smartphones in traveling cars to the server by using the client and server side implementations, and measured upload time through 3G and 4G networks. In the experiment, two cars v_1 and v_2 equipped with iPhone 5s running the implemented software traveled in Nara prefecture, and while traveling, v_1 captured and uploaded 20 ten-second videos with 640×480 pixels resolution through 3G network, and v_2 captured and uploaded 13 ten-second videos with 1280×720 pixels resolution through 4G (LTE) network. NTT DOCOMO's SIM cards are used for 3G and 4G communications. All uploads were done in different places within the same day. The sizes of videos with 640×480 and 1280×720 pixels are about 4.3MB and 14.0MB on average, respectively.

As a result, average upload time and throughput by v_1 (using 3G) were 112.8 seconds and 318.6Kbps, respectively, while those by v_2 (using 4G) were 41.9 seconds and 3.8Mbps, respectively. This time, the throughput using 4G (LTE) network was about 10 times higher than that using 3G. Nevertheless, upload time through both 3G and 4G is within tens to hundreds seconds, and we believe this upload time will not be a problem for practical use.

Performance of image analysis

By using the software of the smartphone side implementation installed on iPhone 5s, we measured the performance of SakuraSensor when applying it to various 1-second videos with 640×480 resolution.

As a result, our implementation of SakuraSensor processed these videos at 11.3 frames per second (fps) on average. Since cherry index does not so quickly change over frames in general, calculating cherry index, for example, every 5 frames is considered to be sufficient. In this case, the current implementation of SakuraSensor can process a video stream (640×480 pixels, 30 fps) in realtime. We believe this result is good enough for practical use.

K-STAGE SENSING METHOD

It is infeasible for a car to sense a long distance alone, because the calculation of cherry intensity consumes much computational resources and requires a lot of communication cost for uploading recorded videos. To mitigate the load of a smartphone, in this section, we propose the k-stage sensing method. In this method, when a car performs sensing, the areas of the nearby PoIs, which are already detected by the preceding cars, are sensed intensively.

In the k-stage sensing method, sensing granularity is controlled by two parameters, i.e., *sensing interval* and *sensing radius*. Sensing interval (in meters) represents how often a car performs sensing, and sensing is performed every time a car runs for a distance specified by the sensing interval. The set of available values of the sensing interval is denoted by $I = \{int_1, ..., int_k\}$, where $int_1 > ... > int_k$. Also, sensing is performed at the specific locations within the sensing radius (in meters) of a circle centered at the existing PoI. The set of the available values of the sensing radius is denoted by $R = \{rad_1, ..., rad_k\}$, where $rad_1 > ... > rad_k$.

In the following, given a car c, we describe how the sensing interval $int(c)$ and the sensing radius $rad(c)$ of the car are determined. First, when c performs sensing, c uploads the information of the current location (GPS data) to the server. Then, the server determines the road that is currently travelled by c, and searches the set of PoIs, denoted by $PoI(c) = \{p_1, p_2, ..., p_m\}$, on the same road, which are detected by the preceding cars within d days past. Here, d is a system parameter to be specified in advance by the service provider depending on the type of scenery (e.g., 7 days in the case of flowing cherries). If $PoI(c) = \emptyset$, then the sensing is performed at the largest granularity, i.e., $int(c) = int_1$ and $rad(c) = rad_1$. Otherwise, the sensing granularity is increased compared with the preceding cars as follows. Let c' be the car that detected a PoI in $PoI(c)$ and performed sensing most recently among the preceding cars, and int_i and rad_i be the sensing interval and radius of c', respectively. Then, we set $int(c)$ and $rad(c)$ to $int(c) = int_{i+1}$ and $rad(c) = rad_{i+1}$, respectively. Eventually, when $int(c')$ and $rad(c')$ reach to int_k and rad_k, respectively, then the sensing granularity will be reset: i.e., $int(c) = int_1$ and $rad(c) = rad_1$.

Using the sensing granularity determined above, sensing of each car is performed as follows, as illustrated in Fig. 9. In

[2]https://nodejs.org/
[3]http://einaros.github.io/ws/
[4]https://github.com/square/SocketRocket

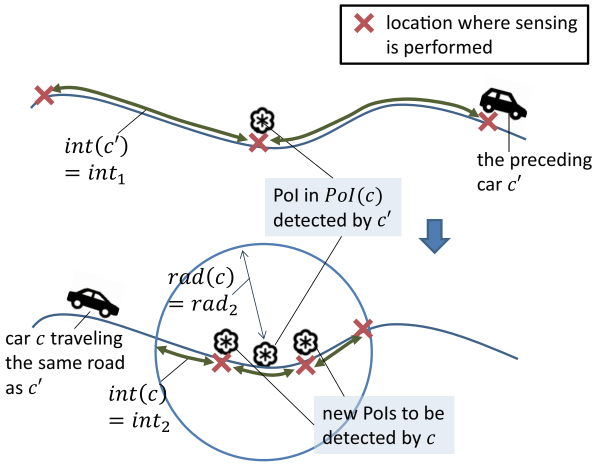


Figure 9. k**-stage sensing method**

the case when $int(c') = int_1$ for a preceding car c', sensing is performed at each location of the car every time after the car c' traveled int_1 meters (upside in Fig. 9). In this case, rad_1 is not used. After that, when a following car c enters the same road as c', it sets its sensing interval and radius to int_2 and rad_2, respectively, because a PoI in $PoI(c)$ is found on the road. Then, c performs sensing at each location every time after c traveled int_2 meters while c is in the circle centered at the PoI with radius rad_2 (downside in Fig. 9).

EVALUATION

In this section, we evaluate SakuraSensor in terms of accuracy of flowering cherry detection based on cherry intensity and effectiveness of k-stage sensing.

Accuracy of Cherry Intensity

The accuracy of cherry intensity is evaluated through the following steps: (1) we recorded a bunch of videos denoted by S by car-mounted smartphone, where flowering cherries show up in some of S; (2) the set of videos denoted by C with flowering cherries were manually selected from S as ground truth data; (3) we classified S into the videos with flowering cherries denoted by C' and those without flowering cherries $(S - C')$ by SakuraSensor; and (4) we compared C and C'.

Table 1. List of scenes

scene name	date	vehicle	area	length (min.)
S_1	Mar. 31	v_1	Aichi Pref.	17
S_2	Apr. 5	v_2	Nara Pref.	12
S_3	Apr. 10	v_2	Nara Pref.	66
S_4	Apr. 10	v_3	Nara Pref.	261
S_5	Apr. 10	v_4	Nara Pref.	186
S_6	Apr. 11	v_1	Gifu Pref.	72
S_7	Apr. 12	v_2	Osaka Pref.	137
S_8	Apr. 18	v_1	Aichi Pref.	89

The set of videos S used for the experiment was recorded by multiple cars while driving along cherry-lined roads in March and April of 2014. There are eight combinations of recording date and car called *scenes* as shown in Table 1. For each scene, we extracted 1-second videos starting at the time randomly selected. All the extracted 1-second videos were manually classified into three classes according to criteria shown in Table 2. Example videos classified to these classes are shown in Fig. 10. The reason why we set 1 second for each video is that it will be difficult to manually classify long videos (more than 1 second) into the classes of Table 2 because the cherry intensity greatly changes among frames in the same video (e.g., cherry intensity could be very high in the first 2 seconds, but there would be almost no cherry in the remaining 8 seconds, when we use 10 second videos). We also considered that different persons might classify the same video to different classes. Thus, in the experiment, two persons independently classified the videos into the classes, and each video which was classified to the same class by the two persons was used as the ground truth data. Table 3 shows the result of manual classification of videos in Table 1.

Table 2. List of classes

class name	criteria
C_1	the ratio of cherry blossoms in the screen is less than 5%
C_2	the ratio of cherry blossoms in the screen is at least 5% and less than 25%
C_3	the ratio of cherry blossoms in the screen is at least 25%

Table 3. Number of videos in each class by manual classification

scene name	C_1	C_2	C_3
S_1	79	17	10
S_2	93	10	17
S_3	372	43	3
S_4	1613	96	45
S_5	1167	6	0
S_6	261	47	72
S_7	888	1	0
S_8	521	10	7
Total	4994	230	154

We mechanically classified 1-second videos into three classes of Table 2 according to the cherry intensity computed by SakuraSensor. To create H-S histogram required for cherry intensity computation, we extracted images with flowering cherries as follows. First, we randomly selected 50 videos from videos which were classified to C_2 or C_3. Here, we selected the videos from as different scenes as possible. Next, in each extracted video, two to five areas showing cherry blossoms were manually extracted (avoiding non-cherry blossoms areas) and stores as image files using the tool shown in Fig. 3. Here, area size is 48×48 pixels. Finally, 148 areas were extracted and stored as image files. We show part of them in Fig. 4.

Then, we divided the set of videos in each class to the training set and the test set. The training set is used for learning mechanical classification of videos to each class based on cherry intensity. More precisely, in the learning process, cherry intensity of each video in the training set is calculated and the median value of cherry intensity values of all the videos in the training set is computed. Let $c.med$ denote the median value of cherry intensity in the training set of class

Figure 10. Example videos of each class

Table 4. Confusion matrix of classification results by SakuraSensor

Actual class \ Predicted class	C_1	C_2	C_3	recall
C_1	2258.8	226.5	11.6	0.90
C_2	29.4	74.3	11.2	0.65
C_3	1.3	11.4	64.3	0.84
precision	0.97	0.24	0.74	

c. When a video m in the test set is given, its cherry intensity $D^{mov}(m)$ is computed and m is classified to the class c' so that $|D^{mov}(m) - c'.med|$ is the smallest.

For class C_1, after shuffling items in the set of videos, the first half and the last half were selected as the training set and the test set, respectively. For classes C_2 and C_3, we similarly divided the set of videos to the training set and the test set, where the videos used for creating H-S histogram of SakuraSensor are not included in the test set. To mitigate the influence by shuffling, we conducted the above process 30 times and averaged the results.

Parameters used for the experiment are as follows. Frame size and the frame rate of each video are 640×480 and 29.97 fps, respectively. The cherry intensity of each video is calculated as the average value of the cherry intensity values of all frames included in the video. We set the fractal dimension threshold to 1.7 and the size of square used for calculating fractal dimension was set to 80×80 pixels. We used OpenCV to implement SakuraSensor and `cvCanny` function for detecting edges in fractal dimension calculation. We set the third and the fourth parameters of `cvCanny` to 100 and 200, respectively. The parameters used in the image analysis are empirically selected based on the preliminary experiments.

Table 4 shows the confusion matrix of classification results by SakuraSensor. In the table, The number of row C_i and column C_j shows the number of videos which actually belong to class C_i and are classified to class C_j (average over 30 trials). Fig. 11(a) shows precision and recall for each class calculated from the result in Table 4. The median values of cherry intensity calculated from the training sets of three classes were 0.00033, 0.00791, and 0.03326, respectively.

Table 4 and Fig. 11(a) show that a good classification result is obtained for class C_1 videos. The result suggests that SakuraSensor can achieve low false positive rate for the roads without flowering cherries (non-scenic routes), because the number of videos of C_1 is relatively large (4994 videos in total) in the experiment. On the other hand, for class C_2 videos, the precision is about 0.2, meaning that the result is not so good. The main reason is that many videos included in class

Table 5. Confusion matrix when classifying to two classes

Actual class \ Predicted class	C_f	C_t
C_f	2589	22.8
C_t	12.7	64.3

precision = 0.74
recall = 0.84

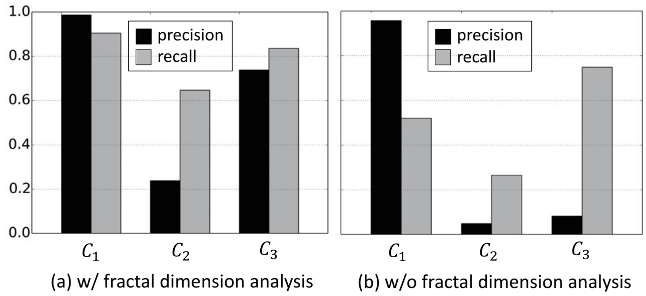


Figure 11. Precision and recall of classification results by SakuraSensor

C_1 were classified to class C_2. To classify videos with small to medium cherry intensity, we need a more accurate classification method. For class C_3 videos, the classification result is good since the recall is more than 0.8 and the precision is about 0.7. However, some C_1 videos were classified to class C_3. We show an example of such a miss-classification in Fig. 12. In this figure, a whity plant (which has similar color to cherry blossoms) shows up in the green and the fractal dimension of the area including the plant is high. In such a case, miss-classification can happen.

To evaluate the effectiveness of fractal dimension analysis, we also derived the classification result when we do not use the filter by the fractal dimension analysis. The precision and the recall in this case are shown in Fig. 11(b). The figure suggests that the classification accuracy totally degrades and especially the accuracy for C_3 is worsen to a great extent when the fractal dimension analysis is not used because many C_1 videos are classified to C_3.

In the afore-mentioned evaluation, we considered the case that intermediate class C_2 exists. In actual use, however, it would be enough to detect videos with only high cherry intensity. Thus, below we consider only two classes: C_f representing videos with low cherry intensity (little cherry blossoms) and C_t representing videos with high cherry intensity (many cherry blossoms). Here, C_1 and C_2 belong to C_f and C_3 belongs to C_t. We show the confusion matrix in this case in Table 5. The precision and the recall of classifying videos to C_t are about 0.74 and about 0.84, respectively. We believe this result is tolerable for the practical use in the early stage deployment of the system.

Accuracy of detecting cherry-lined roads

The evaluation results in previous section were for still image samples randomly picked up from different videos. Here, we try to continuously detect cherry-lined roads while driving. We used the video recorded by the car-mounted smartphone while driving along cherry-lined roads in Gifu prefecture, Japan, in April of 2014. This video is divided to sections

Figure 12. Example of a class C_1 video classified to class C_3

Table 6. Confusion matrix of classification results by SakuraSensor

Actual class \ Predicted class	C_1	C_2	C_3	recall
C_1	3	1	2	0.50
C_2	2	1	1	0.25
C_3	0	1	5	0.83
precision	0.60	0.33	0.63	

at regular intervals with 10 seconds. We evaluated the classification accuracy of all the intervals by the proposed method by comparing with the manual classification results by human as ground truth. Here, classification of each video is done subjectively by one person as follows: when over 50% of frames in the video seem to be in class C_i in Table 2, the video is classified to C_i.

We show the result in Table 6. The percentage of correctly classified sections was 56.3%. The main reason is that many sections included in C_2 were mis-classified to C_1. Then, we investigated the results of classifying to C_t and C_f (videos in C_2 are classified to C_f). Results are shown in Table 7. The percentage of correctly classified sections was 75.0%.

Effectiveness of k-stage sensing

We evaluated the effectiveness of k-stage sensing in the following steps: (1) using the video which is recorded by car-mounted smartphone, where flowering cherries show up; (2) setting the sensing interval to either 100m, 200m, or 500m for all cars (called the *fixed interval sensing* method), and define the parameters I, $R = \{300\text{m}, 150\text{m}, 50\text{m}\}$ for k-stage sensing, where $k = 3$; and (3) comparing the results of sensing times between k-stage sensing and the fixed intervals sensing.

The route used for the experiment is shown in Fig. 13. The number of running cars and the driving speed of each car are defined based on the data of traffic collected by MLIT (Ministry of Land, Infrastructure, Transport and Tourism) Japan shown in Table 8. The sensing start point of each vehicle is shifted randomly between 0 and 600 sec. We defined PoI discovery rate as follows and used it as metric.

$$PoIDiscoveryRate(\%) = \frac{\#DetectedPoIs}{\#AllPoIs} \quad (4)$$

$DetectedPoIs$ correspond to the locations with the same cherry intensity range as class C_3 that are sensed by at least one car, and $AllPoIs$ correspond to the locations with the

Table 7. Confusion matrix when classifying to two classes

Actual class \ Predicted class	C_f	C_t
C_f	7	3
C_t	1	5

precision = 0.63
recall = 0.83

Figure 13. Part of Nara prefectural road 167

same cherry intensity range as C_3 existing in the target area. Figs. 14 and 15 show PoI discovery rate and the sensing times as the number of driving cars increases. Table 9 shows PoI discovery rate for each condition and sensing times per car. This result shows that k-stage sensing has the similar PoI discovery rate to the fixed intervals method with 9.47 times smaller sensing times (about 50 %) than the fixed intervals method.

RELATED WORK

Vehicular Sensing

There are some studies which use in-vehicle sensors or smart phone to collect various information while driving and use the collected information for driving support.

In [10], Eriksson et al. used accelerometer of smart phone for sensing road surface conditions and identifying and sharing damaged areas. In [11], Mohan et al. proposed a method for recognizing road congestion from horn sound collected by microphone of smartphone and road surface condition sensed by accelerometer. In [12], Mathur et al. employed a dedicated ultrasound sensor attached to side of a vehicle for recognizing whether a parking lot is occupied or not. There are some studies which estimate road congestion by WiFi-based localization and GPS logs [13, 14],

SignalGuru [15] is an in-vehicle smart phone application which detects traffic signals through image analysis and shares signal changing timings with other cars through WiFi ad-hoc communication. SignalGuru also navigates a driver to regulate driving speeds so that the car does not have to stop at traffic lights and can save fuels. In [16], You et al. proposed a smart phone application called CarSafe which alerts a driver to dangerous situations such as approaching too close to the car ahead and incautious lane changes by capturing orientation of driver's face and gaze with in-camera and monitoring distance to the car ahead and lane changes with back-camera.

Table 8. Traffic data which are used in simulation

Place	Nara
Traffic volume (7-18 o'clock)	569 [cars]
Running speed	23.5 [km/h]

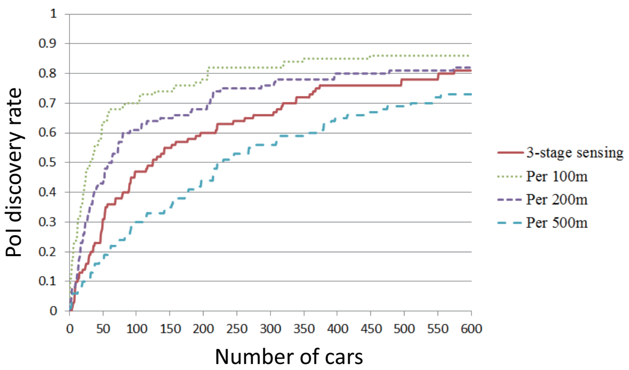


Figure 14. PoI discovery rate

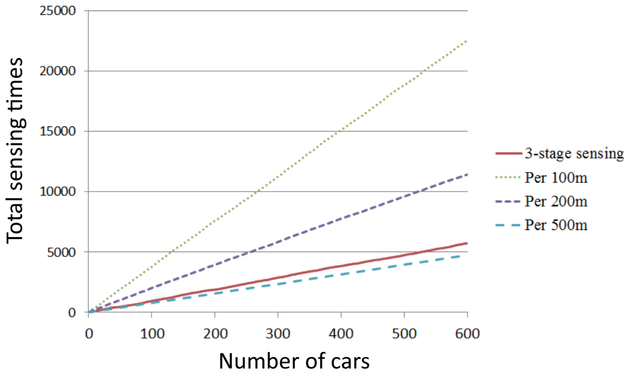


Figure 15. Sensing times

Table 9. PoI discovery rate and sensing times per car

conditions	sensing count to detect all PoI	3-stage sensing	per 100m	per 200m	per 500m
PoI discovery rate (%)	100	81	86	82	73
Sensing times (counts)	824	9.53	37.48	19.00	7.89

Route Recommendation based on Scenery

GPSView [17] automatically extracts scenic spots along roads by detecting spots where many photos are taken from geo-tagged photos uploaded to photo sharing services like Flickr. GPSView also recommends scenic routes to drivers based on the scenic spots information. GPSView improves detection accuracy of scenic spots by picking up only photos taken by travelers, which will be scenic photos at high probability. It also formulates an optimization problem to find the best route taking into account both distance and scenery of routes, and recommends a balanced route to a driver.

In [18], Kawai et al. proposed a system which recommends the best scenic route when a user moves between sight seeing spots by car taking into account scenery along roads. The proposed system identifies PoI on the web by searching a keyword like "scenic spot," sorts the identified PoIs in the order of their scores which are calculated based on the degree of good perspective obtained from 3D map data, and recommends a route including high score PoIs.

Object Detection from Videos Taken While Driving

There are many studies which detect objects from videos taken by cars while driving, for example, detecting cars moving ahead [19, 20], detecting roads [21], detecting pedestrians [22], and detecting signs [23].

To the best of our knowledge, there is little study which detects scenery including flowers, leaves, and trees such as flowering cherries and red leaves from videos recorded by in-vehicle camera while driving. Furthermore, our proposed method is novel in the sense that it does not just detecting objects but also provides a feasible framework for collecting scenery information along roads as well as high-light short videos through participatory sensing.

CONCLUSION

In this paper, we proposed SakuraSensor, a participatory sensing system for automatically collecting and sharing information of cherry-lined roads as well as videos seen from windshield of a car by using car-mounted smartphones. To automatically detect not only existence of flowering cherries but also intensity of cherry blossoms (how densely cherry blossoms exist) from videos, we devised an image analysis method consisting of two techniques which quantify cherry intensity in the video by histogram-based color analysis and region-based fractal dimension analysis. To allow participants to cooperatively find PoIs and save resource consumption by each car, we also proposed k-stage sensing method which dynamically changes sensing interval depending on the distance to already registered PoIs. Our experiments using videos collected while actually traveling cherry-lined roads by several cars showed that our method can classify videos into two classes: videos with little cherry blossoms and those with dense cherry blossoms with precision about 0.74 and recall about 0.84. Also, our k-stage sensing method reduced number of sensing times by each car to about half of the conventional method while keeping the PoI discovery rate.

As part of future work, we will add a mechanism to reduce mis-classification of similar color plants to cherry blossoms by using features of cherry tree, and will conduct more thorough experiments using more videos recorded in various places and conditions. We also plan to extend the system to collect and share other types of information sensed by car-mounted smartphones such as congestion on roads and road-side restaurants and shops.

Acknowledgment

This work was supported in part by JSPS KAKENHI Grant Numbers 25280031 and 26220001.

REFERENCES

1. America's Scenic Byways: `http://scenicbyways.info` (Last accessed: July 17, 2015).

2. NAVITIME: `http://products.navitime.co.jp/function/2519.html` (Last accessed: July 17, 2015).

3. Honda: Internavi LINC: `http://www.honda.co.jp/internavi/service/scenic/` (Last accessed: July 17, 2015).

4. J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava: "Participatory Sensing," In *Proc. of World Sensor Web Workshop (WSW'06)*, 2006.

5. K. Falconer: "Fractal Geometry: Mathematical Foundations and Applications," *John Wiley & Sons*, 1992.

6. M. J. Swain and D. H. Ballard: "Color indexing," *Int'l Journal of Computer Vision*, Vol. 7, No. 1, pp. 11–32, 1991.

7. R. Lopes and N. Betrouni: "Fractal and multifractal analysis: A review," *Medical Image Analysis*, Vol. 14, No. 4, pp. 634–649, 2009.

8. J. Canny: "A Computational Approach to Edge Detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 8, No. 6, pp. 679-698, 1986.

9. I. Fette and A. Melnikov: "The WebSocket Protocol," *RFC 6455*, 2011.

10. J. Eriksson, L. Girod, B. Hull, R. Newton, S. Madden, and H. Balakrishnan: "The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring," In *Proc. of MobiSys'08*, pp. 29–39, 2008.

11. P. Mohan, V. N. Padmanabhan, and R. Ramjee: "Nericell: Rich Monitoring of Road and Traffic Conditions using Mobile Smartphones," In *Proc. of SenSys'08*, pp. 323–336, 2008.

12. S. Mathur, T. Jin, N. Kasturirangan, J. Chandrashekharan, W. Xue, M. Gruteser, and W. Trappe: "ParkNet: Drive-by Sensing of Road-Side Parking Statistics," In *Proc. of MobiSys'10*, pp. 123–136, 2010.

13. J. Yoon, B. Noble, and M. Liu: "Surface Street Traffic Estimation," In *Porc. of MobiSys'07*, pp. 220–232, 2007.

14. A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson: "VTrack: Accurate, Energy-aware Road Traffic Delay Estimation Using Mobile Phones," In *Proc. of SenSys'09*, pp. 85–98, 2009.

15. E. Koukoumidis, L.-S. Peh, and M. Martonosi: "SignalGuru: Leveraging Mobile Phones for Collaborative Traffic Signal Schedule Advisory," In *Proc. of MobiSys'11*, pp. 127–140, 2011.

16. C.-W. You, N. D. Lane, F. Chen, R. Wang, Z. Chen, T. J. Bao, M. Montes-de-Oca, Y. Cheng, M. Lin, L. Torresani, and A. T. Campbell: "CarSafe App: Alerting Drowsy and Distracted Drivers using Dual Cameras on Smartphones," In *Proc. of MobiSys'13*, pp. 13–26, 2013.

17. Y.-T. Zheng, S. Yan, Z.-J. Zha, Y. Li, X. Zhou, T.-S. Chua, and R. Jain: "GPSView: A Scenic Driving Route Planner," *ACM Trans. on Multimedia Computing, Communications and Applications*, Vol. 9, No. 1, Article No. 3, 2013.

18. Y. Kawai, J. Zhang, and H. Kawasaki: "Tour recommendation system based on web information and GIS," In *Proc. of ICME'09*, pp. 990–993, 2009.

19. A. Jazayeri, H. Cai, J. Y. Zheng, and M. Tuceryan: "Vehicle Detection and Tracking in Car Video Based on Motion Model," *IEEE Trans. on Intelligent Transportation Systems*, Vol. 12, No. 2, pp. 583–595, 2011.

20. S. Sivaraman and M. M. Trivedi: "Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis," *IEEE Trans. on Intelligent Transportation Systems*, Vol. 14, No. 4, pp. 1773–1795, 2013.

21. J. M. Alvarez and A. M. Lopez: "Road Detection Based on Illuminant Invariance," *IEEE Trans. on Intelligent Transportation Systems*, Vol. 12, No. 1, pp. 184–193, 2011.

22. D. Geronimo, A. M. Lopez, A. D. Sappa, and T. Graf: "Survey of Pedestrian Detection for Advanced Driver Assistance Systems," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 32, No. 7, pp. 1239–1258, 2010.

23. T. Wu and A. Ranganathan: "A Practical System for Road Marking Detection and Recognition," In *Proc. of IV'12*, pp. 25–30, 2012.