https://doi.org/10.2352/ISSN.2470-1173.2019.8.IMAWM-418

Paint Code Identification Using Mobile Color Detector

Xunyu Pan and Johnathan Tripp
Department of Computer Science and Information Technologies, Frostburg State University, Frostburg, Maryland, USA

Abstract

Automobile manufacturers identify paint used on their vehicles with identification codes. However, as many automobile manufacturers use different and proprietary naming conventions to these codes, it can be challenging for a normal person to find the specific type of paint applied to a vehicle. In this paper, we facilitate this process by developing a portable mobile system to detect paint codes for vehicles. To determine the best matching paint code using the images captured by a mobile device, several practical issues should be examined. First of all, multiple images are captured through the camera viewfinder and the same collection of pixel values are computed collectively to estimate the most accurate color across a short period of time. Second, all pixels in a square region centered on the selected spot are sampled for color estimation to achieve the best match. Finally, in case multiple matches are detected, the closest match should be displayed to the user first, followed by the next closest values. Extensive experiments demonstrate the color detection and matching functionality of our system is robust against varying lighting environments. The scope of this work can be further expanded to other painting industries such as furniture making and construction.

Introduction

Today vehicle body paint has become more and more complex over the years since the first Model-T left the assembly line. Cars these days come in spectacular hues which are as varied as their owners. When a new vehicle is being assembled, the pieces like the metal frame and plastic bumpers come together from factories all around the world. Automobile manufacturers identify paint used on their vehicles with identification codes. A special code contains the information such as the brand, color, and finish of a particular type of paint. Hence each factory may paint the different pieces before they meet and are assembled together into a single vehicle.

However, vehicle owners may periodically need to repair blemishes or even repaint the entire vehicle. As many automobile manufacturers use different and proprietary naming conventions to these codes, it can be challenging for a normal person to determine the specific type of paint applied to a vehicle when these codes are unavailable or difficult to be located. Moreover, the paint color of a vehicle is prone to be affected by numerous interference factors, such as haze, snow, rain, and illumination variation.

To tackle these challenges, we facilitate this color recognition process by developing a portable mobile system to detect manufacturer paint codes for vehicles. This application can be imported to a mobile computing device such as smartphone or PDA with integrated digital camera. More specifically, the system captures images of a user selected region of a target vehicle aiming to estimate the paint color. To determine the best matching paint code using the images captured by a mobile device, several practical issues should be exmined.

First of all, the hand-held mobile device can be used when a user works indoors, or outdoors under various weather conditions. Varying lighting environments can greatly affect the exposure compensation of the camera, resulting in inaccurate color estimation. To address this issue, multiple images are captured through the camera viewfinder and the same collection of pixel values are computed collectively to estimate the most accurate color across a short period of time. Second, the paint color may not be sufficiently estimated using one single pixel at the exact spot selected by a user. Hence all pixels in a square region centered on the spot are sampled for color estimation to achieve the best match. To prevent the background scene from being captured by the camera, the user is instructed not to make a selection along the edges of a vehicle. Finally, the sample paint code colors acquired online are only an approximation of the true paint color. But there are only limited paint colors available for a vehicle in a specific model year. For the best estimate of the paint color, the RGB channels of both the camera input and the true paint of a particular vehicle are compared to identify the closest match along the color spectrum. In case multiple matches are detected, the closest match should be displayed to the user first, followed by the next closest values. For example, the user can determine the appropriate paint code for visually similar colors such as a dark blue and black from the output results of the mobile application. We evaluate the performance of the proposed mobile system indoors and outdoors under various weather conditions.

Extensive experiments demonstrate the paint code identification and color matching functionality of our system is robust against varying lighting environments. Initial implementation of the system stores the dataset of paint codes logically within the application, which can be potentially migrated to the cloud in the near future. The scope of this work can be further expanded to other painting industries such as furniture making and construction.

Related Work

As a notable and stable attribute, vehicle color serves as a useful and reliable cue for vehicle detection. In recent years, vehicle color recognition has received much attention and has been widely used in a variety of applications such as traffic surveillance systems and intelligent vehicles [1, 2]. Most approaches for vehicle detection analyze vehicles from images or videos. Unlike many techniques using background subtraction to extract motion features for vehicle recognition[3, 4, 5], color-based vehicle detection methods [6, 7, 8, 9, 10] rely on global color feature instead of local image features [11, 12] which require full image or video search [13, 14] and are impacted by many environmental factors such as weather and cluttered background.

While there is a rich body of works concerning vehicle detection using color features, very few studies have been made to apply vehicle color recognition for paint code identification. The success rate of vehicle color recognition highly relies on the specific color of a vehicle and the local lighting conditions. Aiming to reduce the effect of lighting change, a color-correction technique [15] was proposed to minimize the color distortions due to various local lighting conditions. However, this technique just roughly classifies vehicles into seven color categories, where paint code identification requires exact color match.

Methods

With the rapid growth in usage of mobile devices and their powerful integrated camera, more and more practical applications have been developed on mobile platforms[16, 17, 18]. We describe in this work a novel color identification system implemented on mobile devices to detect the paint codes for most vehicles. The proposed system is specifically designed for vehicle owners to facilitate the process of paint code identification when automotive paint is required for their vehicles. The purpose of the system is to allow users to determine the manufacturer paint code for their vehicle by pointing their mobile device at the vehicle and tapping on a spot of the vehicle via a live camera viewfinder integrated in the application's main interface. The paint code identification system can be deployed on any mobile device and requires no additional hardware equipment. Users interact with the mobile system through a control panel as shown in Figure 1. When a user touches the screen of the mobile device, a square region around the touch spot is captured for one or more times. Multiple bitmap images from the camera feed are created and the corresponding color information is estimated. The detected color of the vehicle is compared against a list of sample paint colors to determine a matching paint code for that vehicle.

The paint code identification system consists of four major functional components: (a). A vehicle configuration module to allow the user to manually input known information. The information consists not only of vehicle information such as make, year, and model, but also of system setting information such as the detection region size and the number of sample images. The scrolling user interface (UI) elements for region size and other parameters are based on the touch screen technologies supported by most of the today's mobile devices; (b). A color estimation module to detect the color of a vehicle under various weather and lighting conditions. Depending on system configuration, vehicle color is estimated using multiple images for a user selected region of the vehicle; (c). A code searching module to find the candidate paint codes based on user vehicle configuration input. All paint codes and their corresponding sample colors are stored in a local database which is extendable whenever it is necessary; (d). A color matching module to match the estimated vehicle color with existing candidate sample paint colors. This module helps to detect the most similar paint codes based on their color similarity with the estimated vehicle color. Shown in Figure 2 is the high level logic overview of the described vehicle paint code identification system.

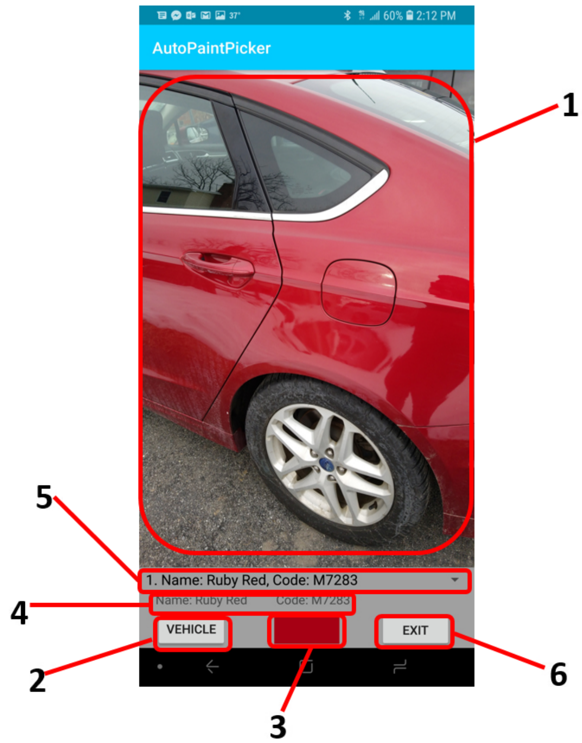

Figure 1. The Graphical User Interface (GUI) of the proposed mobile paint code identification system: (1) A live camera preview from the users mobile device, (2) A button allowing the user to return to the vehicle and system configuration interface at any time, (3) A PaintView *object to display a sample color for an identified paint code, (4) Labels for the common name and manufacturer code of a matched paint color, (5) A spinner view object showing the list of matching paint codes for a target vehicle from most to least likely, and (6) A button to exit the application.*

Vehicle Configuration

The main interface of the proposed system is shown in Figure 1. This interface allows users to see a target vehicle and determine a matching paint code by touching the screen of the mobile device on a selected spot of their vehicle in the camera preview. When this interface is loaded for the first time, system calls are made to the Android Camera APIs to create camera view objects that allow the application to initialize and display the camera feed. The application makes two permission requests for device storage and camera access respectively. Furthermore, the white balance and exposure values for the camera feed are adjusted automatically by the Camera API, no user alteration of settings is required.

The first step for the paint code identification system is for the user to input the information of a vehicle from the initial selection screen that appears when the application starts. The vehicle information can be modified anytime at the vehicle and system configuration interface via a "Vehicle" button on the main interface of the application. As shown in Figure 3, the operations of vehicle and system configuration are completed through this interface. Users are allowed to modify vehicle information such as "Make", "Year", and "Model", as well as system setting such as "Region Size" and "Number of Rounds". Once the user is finished

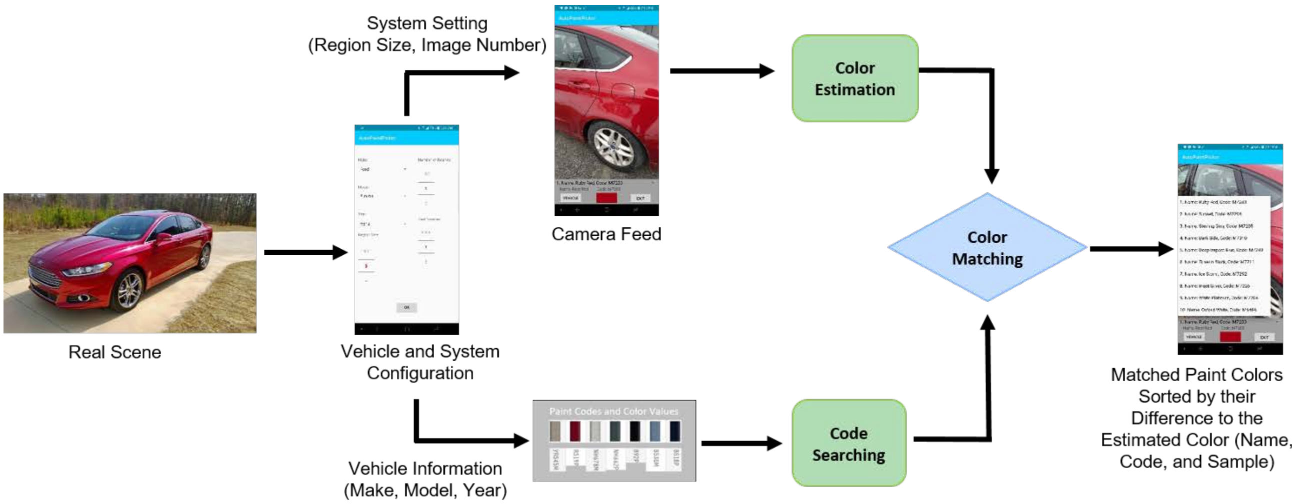

Figure 2. *The overview of the proposed paint code identification system: A user first makes vehicle and system configuration for the proposed color identification system. The system camera then captures real scene images with a target vehicle. The color of the target vehicle is estimated based on one or more captured images. Meanwhile, the paint codes are searched in a local database with user input vehicle configuration. Sample colors of the candidate paint codes are matched with the estimated color of the target vehicle. The best matching paint code associated with its common name is displayed to the user. In addition, the user can view the other possible paint codes for the target vehicle sorted in order from least to greatest difference from the estimated vehicle color.*

configuring the settings for the application, the user can click the "OK" button to be taken to the application's main interface.

Color Estimation

Depending on the system configuration chosen by the user, the camera feed may be sampled one or more times to obtain color information about the target vehicle. When the user touches the screen, the UI thread is temporarily suspended to allow a bitmap image of the camera feed to be captured and stored for later processing. A *MotionEvent* object is also stored when the user touches the screen, and this is held throughout the execution of the algorithm. This *MotionEvent* object holds information about the coordinates of the touchscreen press spot and is required to invoke the *OnTouchListener* that is used to sample the screen. This process is repeated for as many times as the "Number of Rounds" option was set in the configuration settings. The UI thread must be interrupted in order to obtain the bitmap because the information about the camera feed is obtained through the camera preview *OnTouchListener* method, which is called initially when the user touches the screen, and then again through code invocation as many times as needed to capture all of the images.

For each bitmap image captured by the application, a square region of image pixels is stored in an array to act as the sampled color information from the image. Both the width and height of this region is set by the "Region Size" parameter from the configuration screen. The region is centered on the spot where the user touched the screen. All pixels in this region are stored in an array in left to right order, starting at the upper left corner of the region. If the location where the user touched the screen is such that the edges of the region would be out of boundary for the screen, the region is simply offset to keep the region contained within the screen.

When sampling the camera feed, the raw display information from the bitmap image must be converted into a hexadecimal color string in the RGB color space that can be easily read and manipulated. Hence, the integer value of each pixel is formatted into a 6 digit hexadecimal string. The resulting hexadecimal string is stored into an array for later comparison with the appropriate paint color. To obtain the average color for the region, the hexadecimal string of each pixel is broken into its red, green, and blue channels as well as added to each subsequent pixel. The average of each channel is then taken and combined together into a hexadecimal color string representing the average color for the region. This average color of detection region is the estimated color for the target vehicle.

Code Searching

We collect vehicle paint codes from different sources including the Internet. All collected vehicle paint codes are stored in a local database with the paint information: "Common Name", "Manufacturer Code", and "Sample Color" along with the corresponding vehicle information: "Make", "Year", and "Model". After the target vehicle configuration is entered by the user at the vehicle configuration interface, all possible paint codes are searched and located in the database. A list of candidate paint codes is loaded from the local database which is accessible by the application through a JSON file according to the input vehicle configuration. The list of candidate paint codes are evaluated to remove any code whose sample color's RGB components do not fall within an acceptable range of similarity to the estimated color.

Color Matching

With the remaining list of paint codes, each sample paint color is compared to the estimated color of the target vehicle to determine the closest match. More specifically, the sample color of a paint code is compared to the estimated vehicle color by breaking

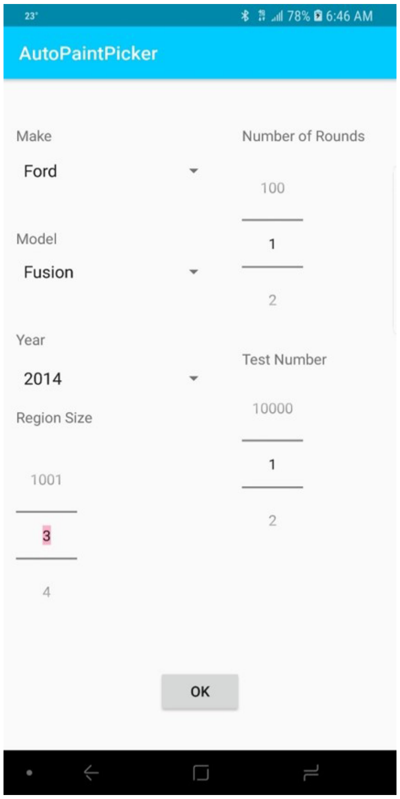

Figure 3. The vehicle and system configuration interface: Users are allowed to modify vehicle information and system setting.

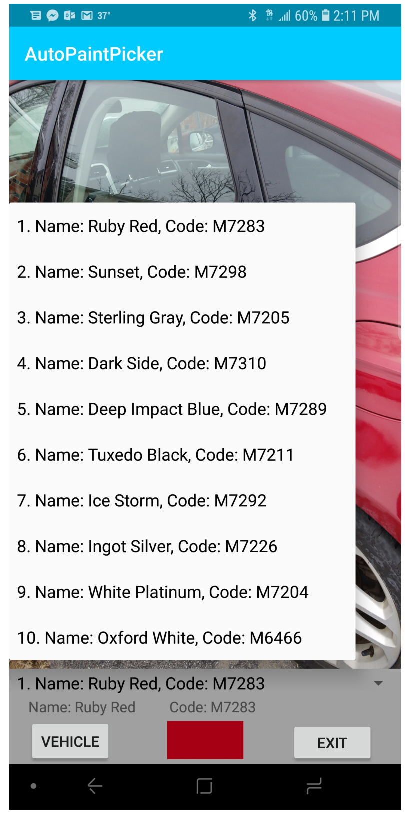

Figure 4. The paint code identification results: Users can view the best matched paint code of the target vehicle and the other possible paint codes sorted in order from least to greatest difference from the estimated vehicle color.

them into their red, green, and blue channels respectively. These results from all three channels are combined together to show the total difference between a sample paint color and the target vehicle color. When the best match has been determined, the most similar paint code is displayed to the user on the main interface. This paint code is shown with the common name and manufacturer code. Meanwhile, the sample color of the best match is also shown on the bottom-center of the interface. In addition, as shown in Figure 4, the user can view the other possible paint codes for the target vehicle sorted in order from least to greatest difference from the estimated vehicle color.

Results

The paint code identification system was written in Java language with the support of the Android Camera APIs. The proposed system was developed using Android Studio 3.0 on a Windows 10 computer. The system was later installed and tested on an Android SDK 26 (8.0 Oreo) platform.

We evaluate the system performance from two perspectives: *Color Estimation Accuracy* and *Code Recognition Rate*. The *Color Estimation Accuracy* measures the average difference between the estimated vehicle color and the sample paint color. The *Code Recognition Rate* measures the average rate for the paint code successfully identified.

Experiments were first conducted using different numbers of images and region sizes. The “Number of Images” parameter was tested using 1, 5, 10, 20, and 50 images respectively, while the “Region Size” parameter was tested using regions of 5, 10, 20, 50, and 100 pixels respectively in the width and height dimension. Each parameter was tested for each vehicle under each lighting condition for a total of 18 individual tests. As shown in Table 1, more estimated images and larger region size typically means better *Color Estimation Accuracy*. Meanwhile, it also means more computation power and time. Hence we choose image number 10 and region size 50 pixels as the default system setting for the following experiments since they help to achieve the best tradeoff between detection performance and time consumption.

With default system setting, we further test the performance of our system on successful paint code recognition in different environmental scenarios. *Code Recognition Rate* highly relies on the specific color of a vehicle and the local lighting conditions. Data collection for our experiments were performed using 3 vehicles. The vehicles tested were a 2014 Ford Fusion (Ruby Red), a 2006 Honda Odyssey (Midnight Blue Pearl), and a 2007 Honda Odyssey (Ocean Mist Metallic). We tested our mobile system in one indoor scenario and 3 outdoor scenarios: Sunny, Cloudy, and Snow. All tasks for outdoor data collection are completed during daytime with normal daylight conditions.

As shown in Table 2, all three vehicles achieve high *Code Recognition Rate* in both indoor and outdoor scenarios. These results demonstrate that the proposed system is very effective to recognize vehicle color and to identify the corresponding paint code. Additionally, we also have two interesting observations. First, the

Table 1. Performance comparison of the proposed system on Color Estimation Accuracy (in percentage) for different region sizes (in pixels) and different number of images respectively.

	Region Size					Image Number				
	5	10	20	50	100	1	5	10	20	50
Ford Fusion (Ruby Red)	86.95	84.79	94.47	94.62	96.18	96.43	94.42	94.57	93.46	96.43
Honda Odyssey (Midnight Blue Pearl)	90.29	88.99	85.49	84.87	92.95	54.85	48.74	90.22	96.86	64.55
Honda Odyssey (Ocean Mist Metallic)	88.86	89.75	95.36	97.36	93.49	88.59	87.79	88.52	88.29	89.63

most successfully identified paint code during testing was for the Ford Fusion. The *Code Recognition Rate* experiments for this car achieved perfect 100% results for all 4 indoor and outdoor scenarios. This may be due to the bold and distinct nature of the vehicles color compared to the other vehicles tested. The bright red paint of the Fusion is more easily distinguished compared to the deep blue of the 2006 Honda, or light blue of the 2007 Honda where those colors may be identified as more black, gray, or white depending on how they are illuminated. Second, strong lighting condition may not be a positive factor for color identification. The system performance for Sunny days is relatively ineffective due to the stronger reflection of light off the paint surface, leading to inaccurate color estimation.

Conclusions

Vehicle color matching is the daunting process of duplicating a color to keep a car looking new and beautiful. While some works have been performed for vehicle detection using color features, the identification of the manufacturer code of paint color of vehicles has not been widely studied. In this work, we describe an effective mobile application developed on the Android platform aiming to identify the paint code of vehicles. The proposed mobile system is convenient for rapid matching of manufacturer paint code with the color of a target vehicle. Our color identification application is portable and user friendly for most vehicle owners who periodically need determine the specific type of paint applied to a vehicle when body painting is required. Experimental results have demonstrated that the proposed system is both efficient and effective to detect the best matching paint code for a target vehicle with default system configuration.

Our mobile paint code identification system has been shown to be convenient and effective in detecting vehicle paint color. Some potential improvements are achievable for the proposed system in the near future: (a). Add automatic vehicle recognition module to avoid user interaction on vehicle selection. (b). Migrate the local data base to the cloud for a larger pool of paint codes. (c). Perform color estimation in other color space such as HSV/HIS to achieve independent color channels and reduce the overexposure problem. (d). Expand to other painting industries such as furniture making and construction. The ultimate goal of the mobile system is to demonstrate the feasibility and the potential for converting any mobile device into a particular tool that can provide real-time paint code identification for various real life objects.

Acknowledgements

This work was partially supported by a Frostburg State University Foundation Opportunity Grant (# 34035).

References

[1] Z. Sun, G. Bebis, and R. Miller, “On-road vehicle detection: A review,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 694–711, 2006.

[2] V. Kastrinaki, M. Zervakis, and K. Kalaitzakis, “A survey of video processing techniques for traffic applications,” *Image and Vision Computing*, vol. 21, no. 4, pp. 359–381, 2003.

[3] S. Gupte, O. Masoud, R. F. K. Martin, and N. Papanikolopoulos, “Detection and classification of vehicles,” *IEEE Transactions Intelligent Transportation Systems*, vol. 3, pp. 37–47, 2002.

[4] G. Foresti, V. Murino, and C. Regazzoni, “Vehicle recognition and tracking from road image sequences,” *IEEE Transactions on Vehicular Technology*, vol. 48, pp. 301 – 318, 01 1999.

[5] R. Cucchiara, M. Piccardi, and P. Mello, “Image analysis and rule-based reasoning for a traffic monitoring system,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, pp. 119–130, June 2000.

[6] C. Hu, X. Bai, L. Qi, P. Chen, G. Xue, and L. Mei, “Vehicle color recognition with spatial pyramid deep learning,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2925–2934, 2015.

[7] P. Chen, X. Bai, and W. Liu, “Vehicle color recognition on urban road by feature context,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, pp. 2340–2346, Oct 2014.

[8] K. van de Sande, T. Gevers, and C. Snoek, “Evaluating color descriptors for object and scene recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 1582–1596, Sep. 2010.

[9] L. Gao, C. Li, T. Fang, and Z. Xiong, “Vehicle detection based on color and edge information,” in *Proceedings of the 5th International Conference on Image Analysis and Recognition*, ICIAR ’08, (Berlin, Heidelberg), pp. 142–150, Springer-Verlag, 2008.

[10] L.-W. Tsai, J.-W. Hsieh, and K.-C. Fan, “Vehicle detection using normalized color and edge map,” *IEEE Transactions on Image Processing*, vol. 16, pp. 850–864, Mar. 2007.

[11] Z. Sun, G. Bebis, and R. Miller, “On-road vehicle detection using gabor filters and support vector machines,” in *2002 14th International Conference on Digital Signal Processing Proceedings. DSP 2002 (Cat. No.02TH8628)*, vol. 2, pp. 1019–1022 vol.2, July 2002.

[12] J. Wu, X. Zhang, and J. Zhou, “Vehicle detection in static road images with pca-and-wavelet-based classifier,” in *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.01TH8585)*, pp. 740–744, Aug 2001.

Table 2. Comparison of the Code Recognition Rate (in percentage) of the proposed system against various lighting environments.

		Outdoor		
	Indoor	Sunny	Cloudy	Snow
Ford Fusion (Ruby Red)	100	100	100	100
Honda Odyssey (Midnight Blue Pearl)	94	88	92	95
Honda Odyssey (Ocean Mist Metallic)	95	87	91	95

[13] X. Pan and S. Lyu, "Region duplication detection using image feature matching," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 5, no. 4, pp. 857–867, 2010.

[14] X. Pan and S. Lyu, "Detecting image region duplication using SIFT features," in *The 35th IEEE International Conference on Acoustics, Speech, and Signal Processing*, (Dallas, TX), 2010.

[15] J.-W. Hsieh, L.-C. Chen, S.-Y. Chen, S.-C. Lin, and D.-Y. Chen, "Vehicle color classification under different lighting conditions through color correction," *IEEE Sensors Journal*, vol. 15, pp. 971–983, 2012.

[16] X. Pan, J. Shipway, and W. Xu, "Learning enhancement with mobile augmented reality," in *IS&T Symposium on Electronic Imaging (IS&T-EI)*, (Burlingame, CA), 2018.

[17] X. Pan and C. Gill, "Multimedia instant messaging with real-time attribute-based encryption," in *IS&T Symposium on Electronic Imaging (IS&T-EI)*, (Burlingame, CA), 2017.

[18] X. Pan and K. Free, "Interactive real-time media streaming with reliable communication," in *SPIE Symposium on Electronic Imaging (SPIE-EI)*, (San Francisco, CA), 2014.

Author Biography

Xunyu Pan received the B.S. degree in Computer Science from Nanjing University, China, in 2000, and the M.S. degree in Artificial Intelligence from the University of Georgia in 2004. He received the Ph.D. degree in Computer Science from the State University of New York at Albany (SUNY Albany) in 2011. From 2000 to 2002, he was an instructor with Department of Computer Science and Technology, Nanjing University, China. In August 2012, he joined the faculty of Frostburg State University (FSU), Maryland, where he is currently an Associate Professor of Computer Science and the Director of Laboratory for Multimedia Communications and Security. Dr. Pan is the recipient of 2011∼2012 SUNY Albany Distinguished Dissertation Award and 2016 FSU Faculty Achievement Award in Teaching. His publications span peer-reviewed conferences, journals, and book chapters in the research fields of multimedia security, image analysis, medical imaging, communication networks, computer vision and machine learning. He is a member of the ACM, IEEE, and SPIE. (Corresponding Author: xpan@frostburg.edu)

Johnathan Tripp received a B.S. degree in Computer Science with Honors from Frostburg State University (FSU) in 2018. He is currently working toward the M.S. degree in Computer Science at FSU.