

Smart Car

Shih-Chia Huang, Department of Electronic Engineering, National Taipei University of Technology, Taipei, Taiwan

Bo-Hao Chen, Department of Computer Science and Engineering, Yuan Ze University, Taoyuan, Taiwan

Sheng-Kai Chou, Department of Electronic Engineering, National Taipei University of Technology, Taipei, Taiwan

Jenq-Neng Hwang, Department of Electrical Engineering, University of Washington, WA, USA

Kuan-Hui Lee, Department of Electrical Engineering, University of Washington, WA, USA

Abstract

In contrast to a traditional mechanical car, the Smart Car is a highly computerized automobile featuring ubiquitous computing, intuitive human-computer interaction and an open application platform. In this paper, we propose an advanced Smart Car demonstration platform with a transparent windshield display and various motion sensors where drivers can manipulate a variety of car-appropriate applications in augmented reality. Similar to smartphones, drivers can customize their Smart Car through free downloads of car-appropriate applications according to their needs. Additionally, three potential car-appropriate applications related to computer vision are investigated and implemented in our platform for increased driving safety. The first and second car-appropriate applications aim to enhance the driving visual field by restoring the low-visibility scenes captured during inclement-weather or nighttime driving conditions to be high-visibility ones, respectively, and display them on a transparent windshield display. We also survey pedestrian tracking techniques that combine multiple driving recorders' information as a mobile surveillance network, including one proposed framework we have developed as the third car-appropriate application. By embedding these car-appropriate applications, the Smart Car has the potential to increase safety of driving conditions both in daytime and nighttime, even in bad weather.

I. INTRODUCTION

UBIQUITOUS sensors, devices, networks, and information are opening the door to a smart world in which smart devices have extended computational intelligence throughout the physical environment to provide reliable, relevant services to users [1]. These devices are getting smarter, more multi-functional, and more customizable to allow users to access and store comprehensive information via many downloadable applications.

In general, the smart devices are characterized by three important properties:

- Ubiquitous computing (ubicomp): assessing information or being assessed interactively and autonomously everywhere and anywhere via various sensors over different wireless protocols.
- Human computer interaction (HCI): the essential interfaces of smart devices that offer an interaction between human and computer.
- Applications platform: allowing users to download the third-party application (app) software for customizing their smart devices.

Thanks to these properties, smart devices can act as critical facilitator for the Internet of Things (IoT) through the use of recent information and communication technologies (ICT), such as mobile operating systems (e.g., Android, iOS, Windows

**Before**

People just used phones to make calls.

**After**

People now use phones to send photos, audio, documents, and video.



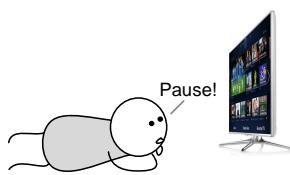
(a)

**Before**

People had access only those television programs that were available at the time through the discretion of broadcasting companies.

**After**

People now have access to on-demand programs and films, as well as internet-based options such as YouTube.



(b)

**Before**

People used cars solely for transport.

**After**

Many cars will offer access to information through transparent windshield displays, built-in GPS, etc.



(c)

Fig. 1: Lives before and after changes in technology for mobile phones, televisions, and cars. (a) Lifestyle brought by smartphones nowadays; (b) Lifestyle brought by smart TVs nowadays; (c) Lifestyle brought by Smart Cars in the future.



Fig. 2: Exterior and interior views of the Smart Car demonstration platform.

Phone, etc.), multimedia interface, internet access (e.g., Bluetooth, NFC, Wi-Fi, 3G, 4G/LTE, etc.), mobile apps, digital cameras, global positioning system (GPS) sensors, motion sensors, and so on.

So what will the next generation devices be like? Let's take a look at the facts in terms of mobile phones and televisions. Mobile phones, which were truly the first pervasive transportable computer, have evolved into multi-service devices now called "Smartphones." The prevalence of these computerized phones allows users to send photos, audio, written documents, and videos to contacts via their phones instead of just using their phones to make calls, as can be seen in Fig. 1(a). In recent years, these modern properties have also been enabling new television (TV) features, turning "dumb" TV into "Smart TV" that give users great potential to watch on-demand videos and access internet-based options via Smart TV's range of online applications (see Fig. 1(b)). Nowadays, the automobile can be regarded as an electronic system in addition to a mechanical one as it incorporates computers [2], [3], informational storage [4], [5], and software [6].

Looking beyond prevalence, we believe that in the near future such advanced properties will include more vehicle technologies, thereby turning "dumb" cars into "Smart Cars." This may include transparent windshield displays (see Fig. 1(c)) to increase driver and passenger safety, convenience, and performance. Hence, the Smart Car should include the following properties:

- Ubicomp for Smart Cars: The in-vehicle system should be networked, autonomously context-aware, and transparently accessible. Additionally, the system should be able to handle multiple datasets and interactions acquired quickly from various sensors.
- HCI for Smart Cars: The windshield should be replaced entirely by a see-through display while embedding various sensors to convey information in a straightforward and easy manner.

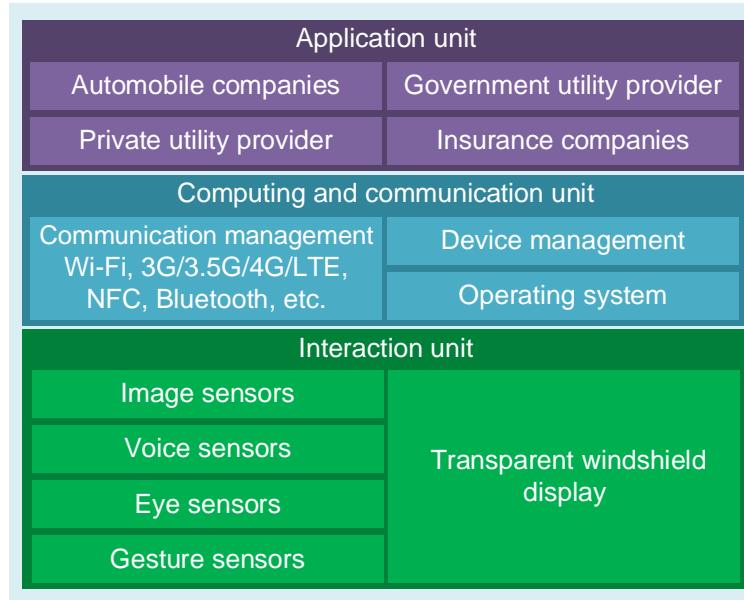


Fig. 3: System architecture of the Smart Car demonstration platform.

- Application platform for Smart Cars: The development platform should be opened up. Hence, several car-appropriate applications from third-party developers would allow users to customize vehicle capabilities and features for their wants and needs.

To illustrate our claim, we present a novel Smart Car demonstration platform (see Fig. 2) equipped with various sensors and a transparent windshield display to receive and display virtual information to driver. The Smart Car that can provide virtual managements of every detail in our lives and navigate everyday traffic in augmented reality for highway, urban streets, and unstructured scenarios, will become a reality in the next few decades. Additionally, we discuss three potential applications, including a visibility restoration application, a nighttime contrast enhancement application, and a driving environment understanding application, to enhance operational safety for Smart Cars when driving in urban street scenarios.

II. SMART CAR PLATFORM

The system architecture of the Smart Car demonstration platform is divided into three major units (see Fig. 3):

- 1) The interaction unit, with a see-through windshield monitor to display content quickly instead of a traditional windshield, and with a variety of in-vehicle sensors to convey information to users or to the computing and communication unit.
- 2) The computing and communication unit, which provides access and execution of both external and internal services with internet access capability.
- 3) The application unit, which facilitates downloading of third-party apps.

A. Interaction unit

In our Smart Car demonstration platform (see Fig. 4), there are two classes of input sensors to convey information about the environment outside the vehicle and the user's intended actions, and one class of output devices used to convey information about the feedback of the computing and communication unit.

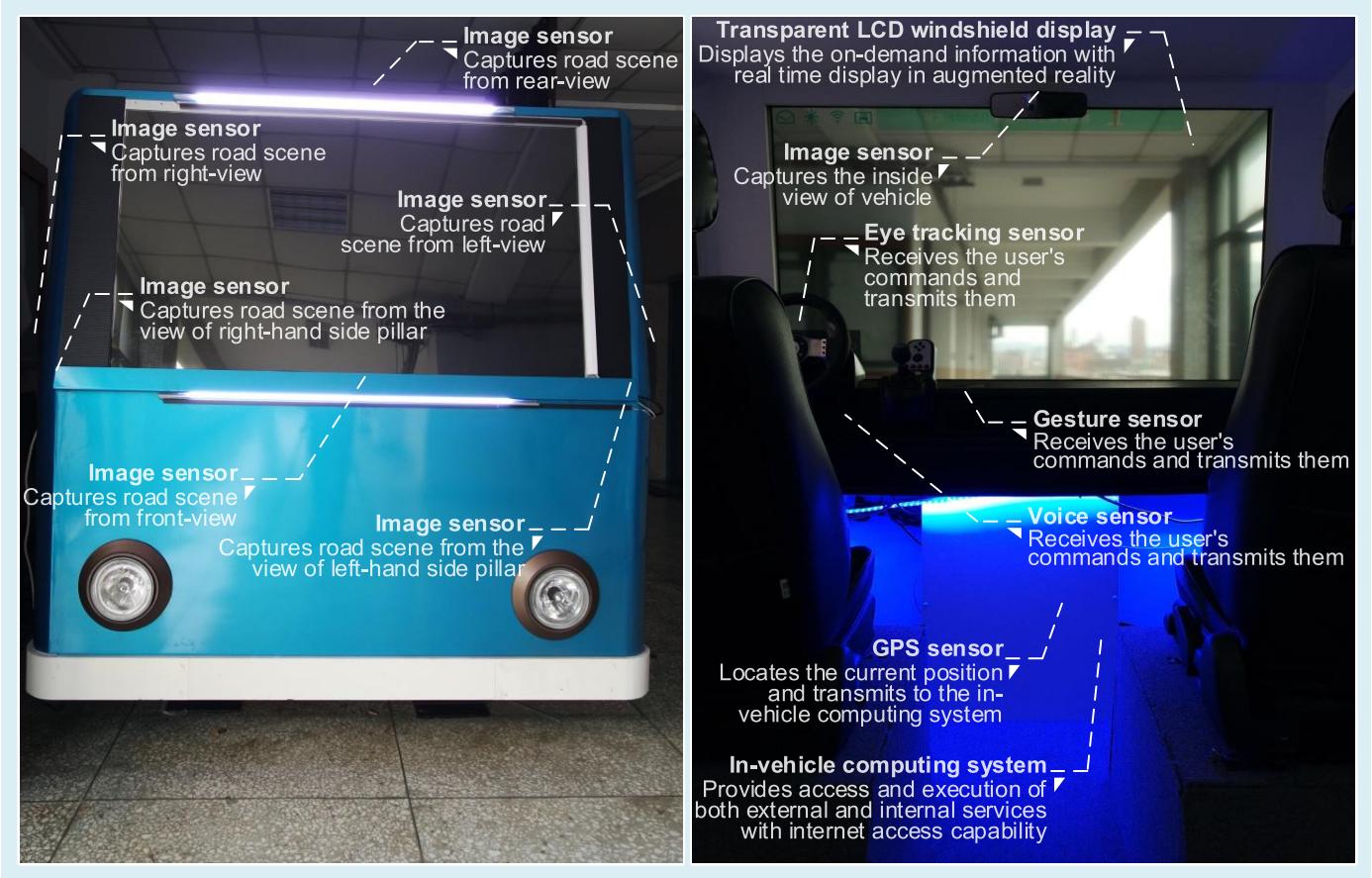


Fig. 4: The Smart Car's configuration.

- ***Input sensors - vehicle surround sensors:*** There are six image sensors plus a GPS sensor around the Smart Car demonstration platform for capturing road scenes and locating the car's position. The configuration of these image sensors, which jointly offer the driver with a 360-degree view out of the vehicle, is given in the left side of Fig. 4.
- ***Input sensors - user behavioral sensors:*** In our Smart Car demonstration platform, there are three kinds of motion sensors (including a gesture sensor, a voice sensor, and an eye-tracking sensor) mounted on the dashboard of the vehicle which receive user's commands and transmit them to the computing and communication unit, as shown in the right side of Fig. 4.
- ***Output devices - transparent windshield display:*** The Smart Car demonstration platform has a transparent display built into its windshield. Hence, the windshield acts as a display that allows users to not only see through the screen but also show on-demand information with real-time display in augmented reality through this visual interface. Here, the on-demand information refers to the driver's requests regarding the content of Smart Car's applications which are made available to driver as needed.

B. Computing and communication unit

The computing and communication unit of the Smart Car should provide real-time access and execution of external/internal services and process multiple data sets acquired from varied sensors of the interaction unit. Hence, the growing complexity

TABLE I: Specific information of in-vehicle computing system.

Item	Features
Embedded Multi-core Processors	<ul style="list-style-type: none"> Processor Base Frequency 2.2 GHz / Max Turbo Frequency 3.1 GHz / Cache 4 MB
Internal Memory	<ul style="list-style-type: none"> Memory Interface DDR3L 1600 / Memory Size 16 GB
Solid State Disk	<ul style="list-style-type: none"> Capacity 1TB / Form Factor 2.5-Inch / Interface SATA III
Communications	<ul style="list-style-type: none"> Bluetooth / WLAN / FM / Transmitter / Receiver (802.11a/b/g/n, Bluetooth V2.1+EDR, 65 nm)
Global Positioning System	<ul style="list-style-type: none"> Tracking Sensitivity-163 dBm / NMEA 0183 data protocol / Time To First Fix / Built-in Super-Cap
Graphics Processing Unit	<ul style="list-style-type: none"> Memory Size 4 GB / Memory Clock 2500 MHz / Memory Interface GDDR5 / CUDA Cores 1536
Power Management	<ul style="list-style-type: none"> Output 19.5 V DC, 9.23 A, 230 W
Mobile Operating Systems	<ul style="list-style-type: none"> Windows 8.1 Enterprise



Fig. 5: Layout combination of the transparent windshield display [45].

of in-vehicle systems requires: integration and cooperation of the software modules; multi-sensor data fusion; stable storage; capability of plug and play. Therefore, the in-vehicle computing system should include specific features, as listed in Table I.

C. Application unit

We gave the Smart Car an open platform where users are able to download applications from automobile companies, government/private utility providers, and insurance companies in order to customize the features, performance, and capabilities.

The applications in this platform can be manipulated through gestures, voice, and eyes. Additionally, the user interface of the transparent windshield display consists of a header container (upper portion of the screen), a content container (mid portion of the screen), and a footer container (lower portion of the screen), as shown in Fig. 5. The header container always displays information such as the status of the car, traffic information, weather conditions, etc. The content and footer containers show the complete and simplified content from the activated application, respectively. The application platform can be downloaded from [7].

TABLE II: Icons of three potential applications for operational safety enhancement in the Smart Car demonstration platform. The first column represents the number of icons and the second column represents three potential applications.

Icon	Application
	Visibility Restoration Application
	Nighttime Contrast Enhancement Application
	Driving Environmental Understanding Application

III. SMART CAR APPLICATIONS

In this section, we implement and discuss three potential applications for Smart Cars related to computer vision which can be used to enhance operational safety. All these three applications were previously developed by the authors of this paper and were adapted to fit into this Smart Car demonstration platform. The icons used for these three applications are given in Table II.

A. Visibility Restoration Application

Inclement weather, such as heavy haze, fog, sandstorms, and so on, can make driving quite difficult and dangerous. For instance, a driver may have trouble seeing the road clearly and need more time to make a decision to brake when driving in heavy haze conditions, as shown in Fig. 6. The ability to use a transparent windshield display that enhances visibility could alert a driver to potential dangers when his/her vision field is diminished in inclement weather conditions.

In general, visibility restoration of a single image can be performed by using the optical model. This model recognizes that the reflected light and distance are linearly correlated with the observed object and camera [8]. The intensity of a hazy image I from the observed object to camera can be represented by direct attenuation (including the intensity of haze-free image J and the intensity of transmission T) and airlight (including the global atmospheric light A):

$$I = JT + A(1 - T) . \quad (1)$$

Let's assume a homogeneous medium and that the intensity of transmission T represents the amount of light reflected back to the camera from the particle surface in which transmission T can be determined as $T = e^{-\beta d}$, where β is the medium attenuation coefficient and d is the distance between the camera and the particle surface. In particular, the aim of visibility restoration algorithms [9]–[11], [13], [14] is to estimate the latent haze-free image J via transmission T from haze image

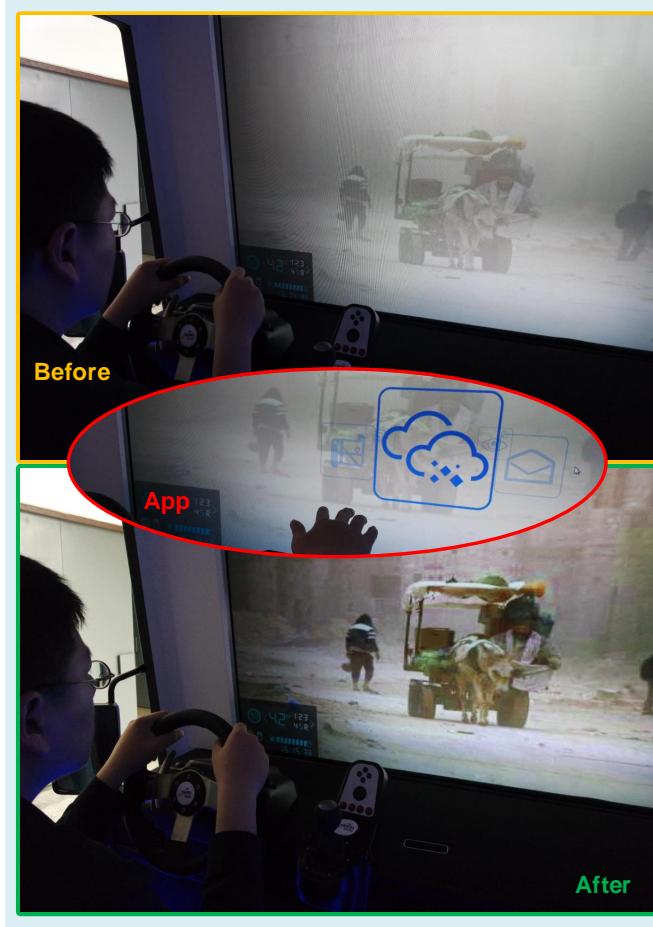


Fig. 6: Manipulation of visibility restoration application in the Smart Car. Upper portion: driving in conditions with poor visibility; mid portion: turning on the visibility restoration application; lower portion: driver's vision field has been improved after turning on the application.

I when there is no additional information available regarding depth and airlight. In practice, some driving situations may appear and disturb these visibility restoration algorithms. Here, we identified the following challenging situations in the field of visibility restoration for driving situations.

- Local lights: When driving in hazy conditions, either the headlights of vehicles or the streetlights are usually turned on to improve the driver's vision field. This often causes misjudgement of airlight for some visibility restoration algorithms, resulting in artifact effects.
- Colorcast problems: It is due to the different kinds of particles that absorb different portions of the color spectrum, and usually results in different distributions of color channels, thereby producing a restored image that still suffers from colorcast problems.
- Gray road: The roadway is usually gray in images captured in the daytime. In foggy conditions, this may result in over-restoration of the road part of the image due to the similarity between the road color and the fog color.
- Deep depth of field: Under hazy weather conditions, the haze is thicker across the vanishing point of the road than it is in front of vehicle (which is close to haze-free). When driving in situations with deep depths of field, the uneven haze density in the image might be misjudged by visibility restoration algorithms, thereby resulting in a restored image that

TABLE III: Challenges and solutions for visibility restoration algorithms in driving conditions. The first column lists the challenges and the second column lists the corresponding solutions.

Challenges	Solutions
Local lights	<ul style="list-style-type: none"> • Hybrid dark channel prior [9]
Colorcast	<ul style="list-style-type: none"> • Gray world assumption [9], White patch-Retinex theory [10]
Gray road	<ul style="list-style-type: none"> • No-black-pixel constraint [11]
Deep depth of field	<ul style="list-style-type: none"> • Bi-histogram modification [12]
Planar surface	<ul style="list-style-type: none"> • Flat-world assumption [13]
Complex architecture	<ul style="list-style-type: none"> • Nonlinear filtering [14]

still appears hazy.

- Planar surface: Hazy images captured on roadways during driving conditions are generally assumed to be planar. As such, some visibility restoration algorithms are not able to effectively restore visibility in hazy images that were not taken along a planar surface.
- Complex architecture: Complex architecture may exist along a road, such as pylons, streetlamps, etc. A restored image can easily suffer from halo effects along the edges of these structures.

Table III gives an overview of common problematic issues. In this article, we take a cue from reference [9] when considering the frequent appearance of local lights and colorcast problems during driving conditions. In the literature [9], a haze removal approach was developed and consisted of a hybrid dark channel prior (HDCP) module, a color analysis (CA) module, and a visibility recovery (VR) module. In the HDCP module, a hybrid transmission map was produced to estimate the haze density while avoiding the misjudgement of local lights as atmospheric lights. The hybrid transmission map can be expressed as

$$t_h = 1 - \frac{\omega\alpha}{\alpha + \beta} d_\Omega(I) - \frac{\omega\beta}{\alpha + \beta} d_\mu(I), \quad (2)$$

where ω , α , and β are the constant factors and can be acquired according to [9]; $d_\Omega(I)$ and $d_\mu(I)$ are the processes of dark channel prior on each color channel of haze image I using patches of sizes 3×3 and 45×45 , respectively. Moreover, the colorcast problem was suppressed by the CA module, where the color adjustment γ_c for c th color channel was measured as

$$\gamma_c = \frac{m_r(I)}{m_c(I)}, \quad (3)$$

where $m_r(I)$ and $m_c(I)$ are the average processes on the R color channel and c th color channel in RGB color space, respectively.

Next, the VR module restored the haze image to haze-free one J_c by

$$J_c = \frac{\gamma_c(I_c - A_c)}{\max(t_h, t_0)} + A_c + \sigma_c(\gamma_c - 1), \quad (4)$$

where σ_c is the gain factor for c th color channel and can be produced by

$$\sigma_r = \underset{\forall l}{\operatorname{argmax}} \operatorname{PMF}(I_r), \quad (5)$$

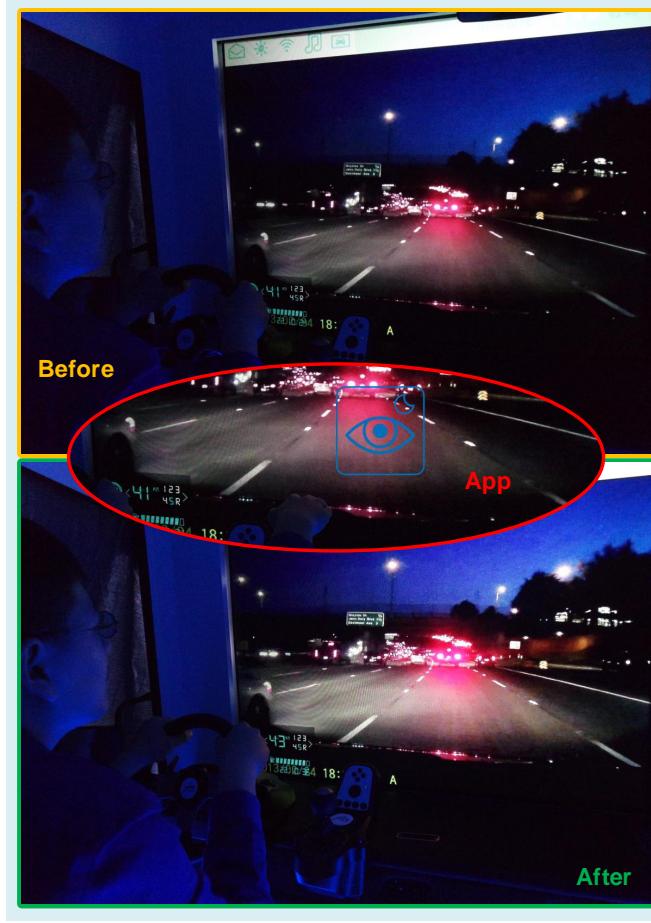


Fig. 7: Driver performing the nighttime contrast enhancement application where the driver's vision field is increased. Upper and lower portions represent the driver's vision field before and after using the application.

$$\sigma_g = \frac{1}{2} \left(\underset{\forall l}{\operatorname{argmax}} \text{PMF}(I_r) + \underset{\forall l}{\operatorname{argmax}} \text{PMF}(I_g) \right) , \quad (6)$$

$$\sigma_b = \frac{1}{2} \left(\underset{\forall l}{\operatorname{argmax}} \text{PMF}(I_g) + \underset{\forall l}{\operatorname{argmax}} \text{PMF}(I_b) \right) , \quad (7)$$

and l is the intensity level. Note that $\text{PMF}(\cdot)$ is the process of probability mass function and A_c is the atmospheric light for c th color channel obtained from d_μ [9]. In addition, the computational complexity of the haze removal approach is principally embodied in its three modules. The complexity of HDCP module is $O(kMN)$ to produce the hybrid transmission map for M row and N column via the patch size of containing k pixels. The complexities of both CA and VR modules are $O(MN)$. After turning on the visibility restoration application (see Fig. 6), the vision field of the driver is improved, such that the driver can see the surrounding environment more accurately and engage in better decision-making.

B. Nighttime Contrast Enhancement Application

At night, the nighttime contrast enhancement application combines enhanced night vision with the transparent windshield display to allow drivers to both identify and highlight the location of threats (e.g., pothole or bump), as shown in Fig. 7.

In general, the current image I captured by using imaging device consists of both the light reflectance R and the visible-light

illumination L according to the literature [15]. This can be expressed as

$$I = LR . \quad (8)$$

The light reflectance R would not be changed due to the illumination variance in the same scene. When the imaging device can receive sufficient illumination, the ideal illumination image I_i can be obtained by the ideal illumination L_i . On the contrary, poor illumination L_p can produce a restored image with poor illumination I_p .

The main aim of various image contrast enhancement algorithms [16]–[25] is to increase the illumination level via the processing function of contrast enhancement $f(\cdot)$, while preserving the image features. This can be expressed as follows:

$$I_i \leftarrow f(I_p) . \quad (9)$$

Various algorithms have been developed to achieve the processing function of contrast enhancement. However, the use of these algorithms during nighttime driving conditions should specifically address the following issues:

- Artifact tolerance: when driving at night, there are various light sources along the roadway to enhance a driver's vision field. The image contrast enhancement algorithms should avoid introducing certain artifacts (also called "blocking effects") as the vehicle is moving.
- Uniform contrast: the image contrast enhancement algorithms should be able to simultaneously and effectively enhance all portions of the current image during night-driving.
- Brightness preservation: after the contrast of the current image is enhanced, the enhanced image should not lose the brightness of various light sources along the roadway.

Hence, several approaches have been proposed to overcome these issues, as listed in Table IV. In this article, we implement the nighttime contrast enhancement application by using algorithm [25] for brightness preservation and artifact tolerance. In the literature [25], the adaptive gamma correction with weighting distribution (AGCWD) was proposed to enhance the image contrast for visual quality improvement. First, the transformation curve T of adaptive gamma correction was devised as follows:

$$T(l) = l_{\max} (l/l_{\max})^\gamma , \quad (10)$$

where l and l_{\max} are each intensity level and maximum level for an 8-bit pixel on V channel of HSV color model in a given image I , respectively. γ is the adaptive parameter for each level, which can be defined as

$$\gamma = 1 - \text{CDF}(l) , \quad (11)$$

where $\text{CDF}(\cdot)$ is the process of cumulative distribution function and can be expressed as follows:

$$\text{CDF}(l) = \sum_{l=0}^{l_{\max}} W(l) / \sum W , \quad (12)$$

TABLE IV: Challenges and solutions for image contrast enhancement algorithms when driving at night. The first column lists the challenges and the second column lists the corresponding solutions.

Challenges	Solutions
Artifact tolerance	<ul style="list-style-type: none"> • Recursive sub-image histogram equalization [16], Optimum recursion based histogram separation [17], Local histogram equalization [18]
Uniform contrast	<ul style="list-style-type: none"> • Spatial entropy-based contrast enhancement [19], Centre-surround Retinex contrast enhancement [20], Gaussian mixture model histogram modification [21]
Brightness preservation	<ul style="list-style-type: none"> • Brightness-preserving bi-histogram equalization [22], Dualistic sub-image histogram equalization [23], Bilateral-Bezier-Curve based histogram modification [24], Adaptive gamma correction with weighting distribution based histogram modification [25]

and $\sum W$ is the sum of process of weighting distribution function W , which can be expressed as follows:

$$\sum W = \sum_{l=0}^{l_{\max}} W(l). \quad (13)$$

Next, the weighting distribution function W can be expressed as follows:

$$W(l) = \text{PDF}_{\max} \left(\frac{\text{PDF}(l) - \text{PDF}_{\min}}{\text{PDF}_{\max} - \text{PDF}_{\min}} \right)^{\alpha}, \quad (14)$$

where α is an adjusted parameter [25], $\text{PDF}(\cdot)$ is the process of probability density function, and PDF_{\max} and PDF_{\min} are the maximum and minimum values of the statistical histogram in $\text{PDF}(\forall l)$, respectively. By using the transformation curve to the dimmed image in the application, the enhanced image potentially provides richer road information to the Smart Car's driver, as can be seen in Fig. 7. The overall efficacy of above processes on computational complexity is $O(MN)$, where M and N are the number of rows and columns in a given image I , respectively. Therefore, a temporal-based entropy model was suggested in [25] and employed to reduce the computational complexity for the visibility restoration application and nighttime contrast enhancement application. For t th frame, the entropy E_t can be measured as follows:

$$E_t = - \sum_{l=0}^{l_{\max}} \text{PDF}(l) \log (\text{PDF}(l)). \quad (15)$$

Here, the first incoming frame is utilized to produce the hybrid transmission map and the transformation curve for these two applications at the beginning of their processes. This information will be recalculated only when the absolute difference between E_t and E_{t-1} is higher than a predefined threshold δ , where δ can be set as 0.05 according to [25].

C. Driving Environmental Understanding Application

Thanks to increasing development of autonomous technologies, many video analytics applications are being applied to moving platforms such as Smart Cars. Human tracking is quite challenging since humans may vary greatly in appearance

due to different viewing perspectives, non-rigid deformations, intra-class variability in shape, and other visual properties. The challenge increases when moving cameras (such as the car recorders) are employed, due to the effects of egomotion, blur, and the issues mentioned above. The introduction of a moving camera invalidates many effective moving object tracking techniques used in static camera, such as background subtraction and a constant ground plane assumption, thus making the task more difficult. Rather than using background modeling-based methods to extract captured human objects, human detectors are widely used to detect images of people in the video frame. Therefore, the challenge is to successfully detect human images captured by moving cameras, and then apply the tracking techniques to those detected, resulting in so-called *tracking-by-detection* schemes.

By applying a human detector to each frame of a video sequence, the tracking scheme becomes a task that associates the detected human objects with each other frame-by-frame. In general, human detection follows two basic steps [26]: foreground segmentation and object classification.

After human object detection, a tracking framework is applied to the detected objects. Previous research of human tracking with moving cameras has involved Kalman filters [27], [28], Particle filters [29], [30], and kernel-based tracking [31], [32], which are widely used in tracking and independently consider each tracked human within the temporal frames. Some studies adopt Multi-Hypothesis Tracking (MHT) [33], [34] and Joint Probabilistic Data Association Filters (JPDAFs) [35] to optimize detected target associations by jointly considering all tracked targets' information over several time steps.

Alternatively, several approaches based on the structure-from-motion (SfM) framework have been developed. The SfM framework, when combined with visual simultaneous localization and mapping (V-SLAM), can be used to calibrate and localize the 3-D positions of the moving camera, and eventually detect, track, and reconstruct moving objects with respect to the static background. This results in a so-called dynamic scene reconstruction. The advantage of the SfM-based approaches is that they locate the objects in 3-D space, so as to better deal with occlusion issues during tracking. Table V summarizes several existing human tracking-by-detection schemes based on a moving camera, with or without 3-D inference [36]–[44].

A robust moving-platform-based human tracking system [45], which takes advantage of the tracking-by-detection scheme and successfully integrates V-SLAM, human detection, ground plane estimation, and kernel-based tracking techniques, is adopted in our Smart Car system. As shown in Fig. 8, this system starts with human detection and V-SLAM for camera calibration. Then, the ground planes are estimated based on the camera motions, so that the 3-D locations of the pedestrians (relative to the cameras) can be inferred. Subsequently, the absolute 3-D locations of tracked humans will become available since the GPS location information of the camera is also available. By taking 3-D information into account, the tracking problem can be reformulated as a constrained 3-D multiple-kernel (CMK) tracking problem [32], which can effectively resolve occlusions during tracking by globally optimizing the data association between consecutive frames.

In [32], the CMK tracking scheme tracks video objects in 2-D space (image), i.e., $x \in \Re^{2 \times N_k}$ in Eq. (16). To efficiently integrate the depth information into the CMK framework, we need to reformulate the problem. First we extend the Eq. (16) from 2-D to 3-D space

$$J^i(X) = \sum_{k=1}^{N_i} w_k^i J_k^i(X), \quad X \in \Re^{3 \times N_i}. \quad (16)$$

TABLE V: Several existing human tracking-by-detection schemes based on a moving camera, with or without 3-D inference.

Tracking Taxonomy	Challenges	Solutions
2-D-based individual object independent tracking	Local optimum data association	<ul style="list-style-type: none"> • Multi-body parts tracking by combining an implicit shape model (ISM) detector and a stereo-odometry-based tracker [32]. • A part-based human detector with a Gaussian process latent variable model to compute the temporal consistency of detections over time [33]. • A color model and the event cone – the time-space volume in which the trajectory of a tracked object is sought in 3-D space [34].
2-D-based multiple object joint tracking	Global optimum data association	<ul style="list-style-type: none"> • A maximum-a-posteriori (MAP) data association formulation of a cost-flow network with a non-overlap constraint on trajectories [35]. • A cost function with the objects' birth and death states, solved by a greedy algorithm [36]. • A coupling formulation to avoid the problem of error propagation to overcome the partial or complete occlusions [37]. • Incorporates constraints of piecewise constant-velocity path smoothness based on the flow network framework [38].
SfM based 3-D inference tracking	Track moving objects within a static background with a moving camera	<ul style="list-style-type: none"> • An incremental V-SLAM system that allows choosing between full 3-D reconstruction or simply tracking moving objects [39]. • Estimate the ground plane by using sparse features, dense inter-frame stereo and object detection based on a real-time monocular SfM framework [40].

This equation is regarded as the local optimization for each individual target i with multiple (N_i) kernels, each of them is weighted by w_k^i . Second, considering the depth information, we assign the visibility of each target as a weight v_i to deal with the global optimization. In other words, the total cost function becomes:

$$J(X) = \sum_{i=1}^M v_i J^i(X) = \sum_{i=1}^M v_i^i \sum_{k=1}^{N_i} w_k^i J_k^i(X), \quad X \in \Re^{3 \times M \times N_i}, \quad (17)$$

where M is the number of the targets in one video frame, and $X = [(X_1^1)^T, \dots, (X_k^i)^T]^T$ is for the i th target and the k th kernel.

Necessarily, the constraint functions $C(X) = 0$ must be considered to maintain the relative locations of the kernels. In [32], two-kernel and four-kernel layouts are proposed to describe a human. Unlike the constraints used in [32], which are mainly based on 2-D geometry, we set the constraints based on 3-D geometry [45]. Hence, for each target i , the movement vector δx can now be iteratively solved by using the projected gradient method [32]. The computational complexity is relatively high, due to the use of human detector on every frame of video, as well as the ground plane estimation. The CMK tracking in

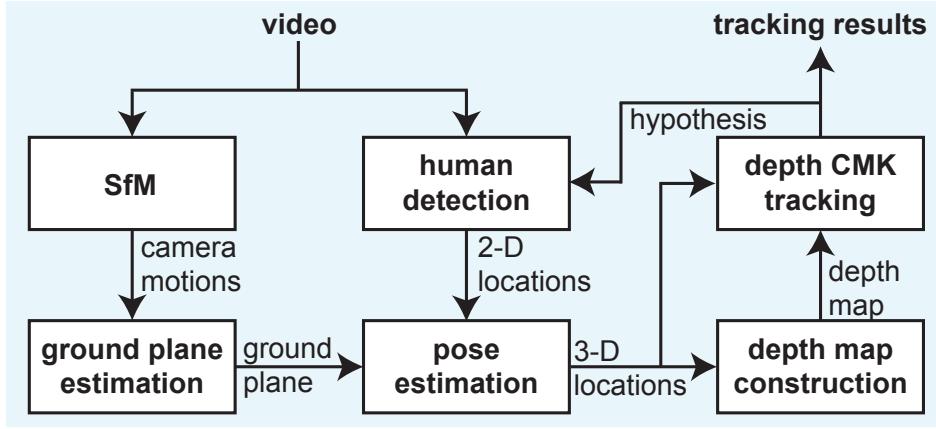


Fig. 8: A moving-platform-based human tracking system [47].

3-D space based on projected gradient can be very fast. At this moment, using a high-end desktop CPU still requires couple seconds to complete one frame of video. This complexity can be relieved if the cloud computing with GPU speedup, real-time processing can be expected.

As examples, representative performances of human tracking for videos obtained from four separate car recorders are shown in the left column of Fig. 9. Moreover, a 3-D visualization of dynamic on-road scenes can also be reconstructed. Its purpose is not only to visualize the pedestrians' paths and movements in a 3-D environment, but also to avoid issues of privacy invasion by using avatar-like 3-D models. When effectively integrated with a 3-D map service, such as Google Earth, we can treat this new 3-D augmented reality visualization as a “dynamic 3-D GPS navigation” system as shown in the middle column of Fig. 9. When there is an event, such as road crossing or activity, we can see the dynamic scene in different aspects of views (see the right column of Fig. 9), which can be effectively displayed within the Smart Car to facilitate drivers' viewing experience.

Moreover, due to the video analytics on each individual Smart Car, it is highly desirable to collectively combine analytics information, such as on-road pedestrian tracking, from multiple nearby vehicles to dynamically facilitate better monitoring of the on-road situation and cooperatively share the information. When tracking pedestrians in a Smart Car with a single moving camera, the camera's field of view (FOV) is limited to the front of the vehicle and dynamically changes when the vehicle moves. To achieve a more complete understanding of the car's surroundings, the tracking must be implemented across multiple moving cameras based on the results of the tracking in a single moving camera. The challenge is that appearance features extracted from the same pedestrian by different cameras may be inconsistent. There are two scenarios associated with tracking across multiple moving cameras: overlapping and non-overlapping FOV scenarios. In an overlapping FOV scenario, a pedestrian simultaneously appears in two or more different cameras' FOVs; therefore, re-identification issues can be facilitated by exploiting the cameras' relationship based on the overlapping views, so as to obtain consistent features from different cameras. In a non-overlapping FOV scenario, a pedestrian enters a camera's FOV then leaves, and later enters other cameras' FOVs, which do not share a common region with the previous FOV. The problem is more difficult due to dynamic environment, lighting changing and unexpected camera locations. Hence, the methods for tracking across multiple static cameras [46] cannot be applied to the tracking across moving cameras. Moreover, both overlapping and non-overlapping cases may occur alternatively during tracking, thus making this task even more challenging.



Fig. 9: 3-D visualization of the scene recorded by four driving recorders. Each row belongs to one driving recorder; the leftmost shows the video frames, the middle shows the corresponding view of 3-D visualization, and the right shows the scene visualized from different aspects [47].

To effectively deal with tracking across multiple driving recorders, we developed a framework [47] to track on-road pedestrians recorded in the videos, where a cloud server is used to collect driving information of several nearby vehicles via a mobile surveillance network. First, pedestrian tracking in a single moving camera is applied to each video. Based on the single-camera-tracking results, we treat the problem of tracking across cameras as a multi-label classification task, which determines each target belonging to one or several cameras' FOVs by considering association likelihood of the target as calculated based on targets' motion cues and appearance features. When a target is out of a camera's FOVs, we predict the target's locations via an open map service such as Google Maps. Moreover, by using the Google Earth service, a 3-D visualization of a dynamic scene can be reconstructed to allow users to see a holistic view or different viewing perspectives of a 3-D scene reconstructed by multiple videos.

Figure 10 shows the system of the developed framework. First, pedestrian tracking, which produces the moving trajectory and associated features of the tracked person (tracklet) in 3-D space, in a single camera is applied to each video. The videos are

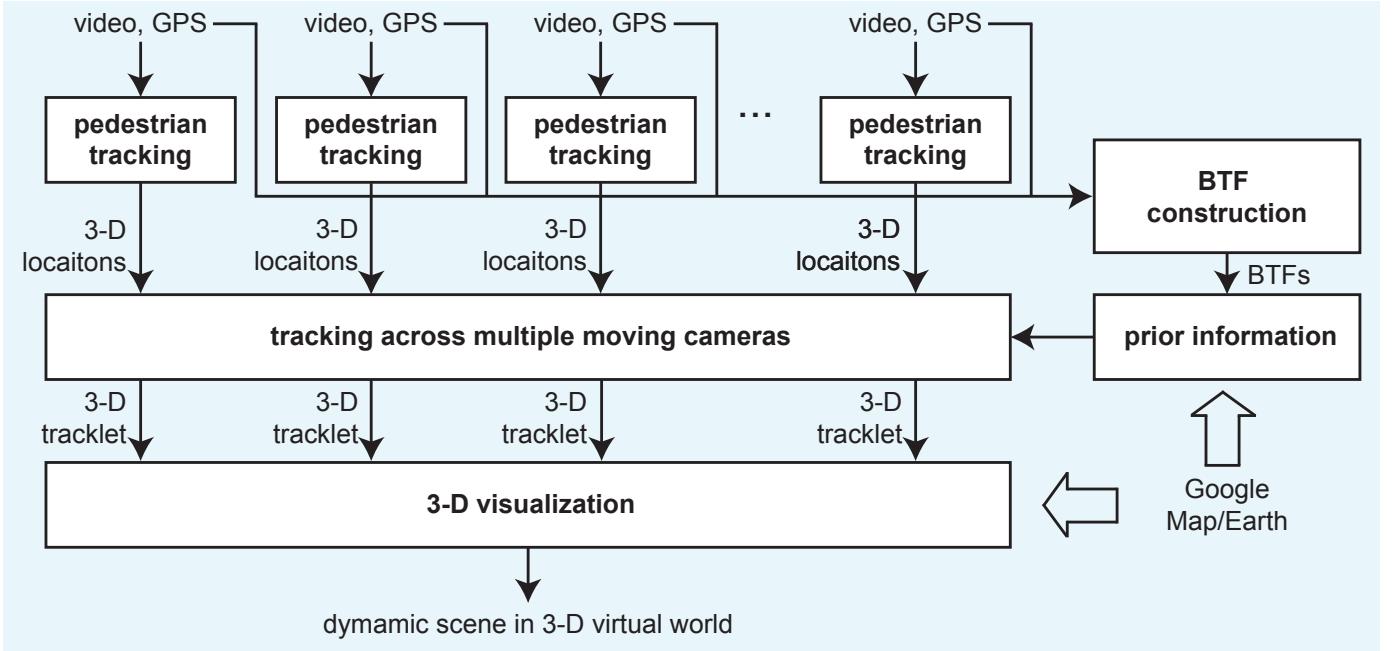


Fig. 10: A system of human tracking across multiple moving car cameras [47].

then used to build Brightness Transfer Functions (BTFs) for compensating the color diversity of the cameras. After estimating the pedestrians' 3-D tracklets in each camera, the pedestrian tracking across multiple cameras is further applied, facilitated with the BTFs and map prior. Finally, the pedestrians' 3-D tracklets are summarized and visualized in the 3-D real-world environment based on an open map service like Google Earth.

As shown in Fig. 11, the pedestrians (in yellow and green bounding boxes) appearing in Camera 1's FOV are successfully tracked (in the same color boxes) later in Camera 3's FOV.

IV. CONCLUSIONS

We proposed the demonstration platform for a Smart Car and further implemented three potential applications on this platform. To implement the Smart Car demonstration platform, we included three important properties. First, the Smart Car is networked and is able to process multiple datasets from various sensors based on the ubicomp property. Next, due to the use of the HCI property, the traditional windshield is entirely replaced on the Smart Car by a transparent windshield display on which the driver can see graphic information and interact with the car via embedded motion sensors. Finally, the application platform of the Smart Car is designed based on the platforms of smartphone and smart TV, is opened up for third party developers, and can be downloaded from [7]. This Smart Car demonstration platform was tested on three proposed applications. Additionally, we have demonstrated the potential capability of a Smart Car to effectively implement and run these three applications during driving conditions. We believe that in the future there will be a great diversity of car-appropriate applications released on the application platform. At that point, the smart world's drivers can build their own Smart Car based on their wants and needs. Consequently, the Smart Car will provide not only a more interesting driving environment, but also a safer one.

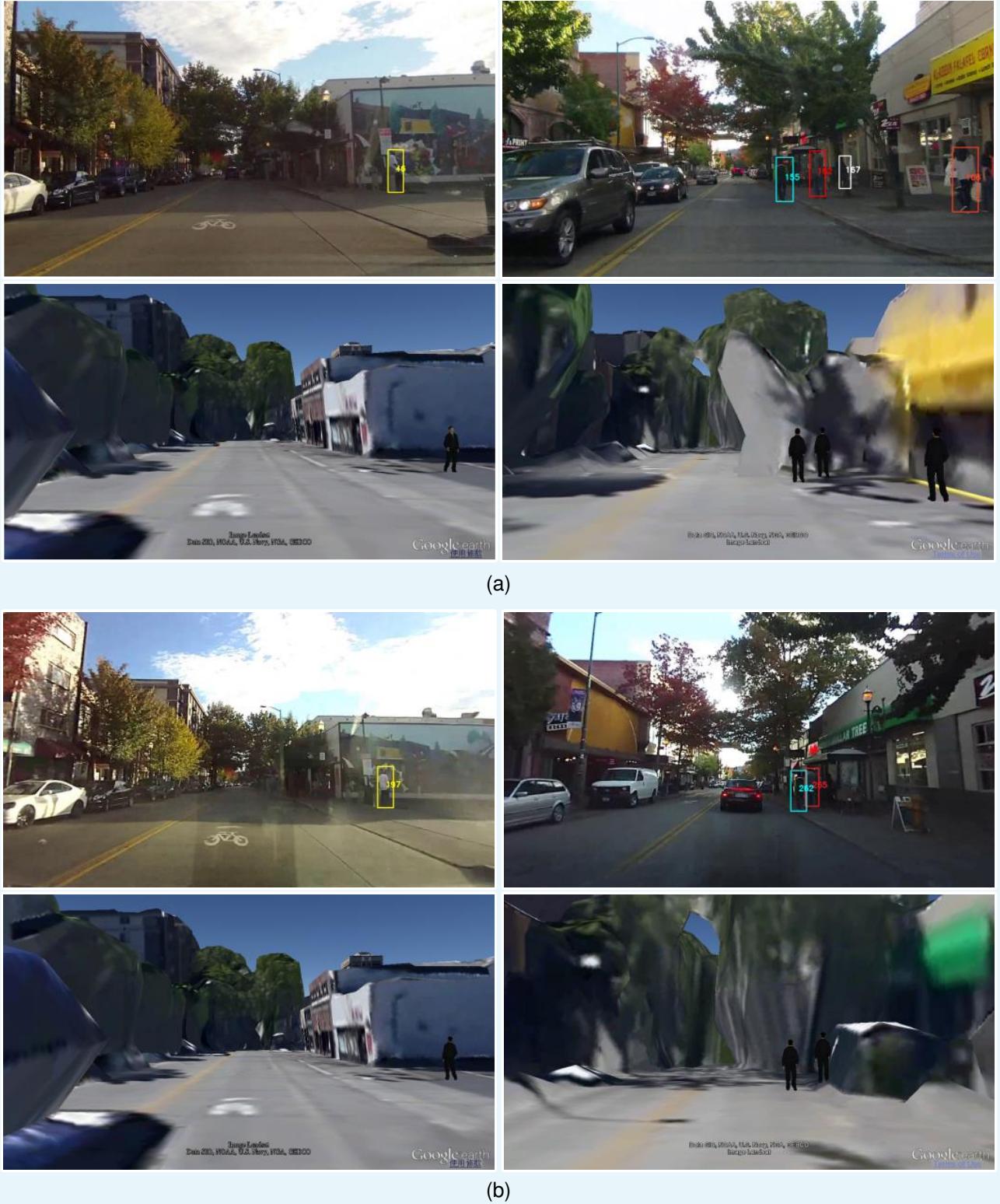


Fig. 11: Visual tracking results, where the top rows are the recorded frames, and bottom rows are the corresponding 3-D visualization [47]. (a) Two human tracked frames from driving recorder-1. (b) Two human tracked frames from driving recorder-3.

V. ACKNOWLEDGEMENTS

This work was supported by the Ministry of Science and Technology, Taiwan, under Grant Nos. MOST 105-2923-E-027-001-MY3, MOST 103-2221-E-027-031-MY2, MOST 103-2221-E-027-030-MY2, MOST 103-2923-E-002-011-MY3 and

MOST 104-2221-E-027-020.

REFERENCES

- [1] S. Poslad, *Ubiquitous computing smart devices, environments and interactions*, New York: Wiley, 2009.
- [2] C. T. Lin, L. W. Ko, and T. K. Shen, "Computational intelligent brain computer interaction and its applications on driving cognition," *IEEE Computational Intelligence Magazine*, vol. 4, no. 4, pp. 32–46, Nov. 2009.
- [3] P. G. Balaji and D. Srinivasan, "Multi-agent system in urban traffic signal control," *IEEE Computational Intelligence Magazine*, vol. 5, no. 4, pp. 43–51, Nov. 2010.
- [4] B. Lin and C. Wu, "Mathematical modelling of the human cognitive system in two serial processing stages with its applications in adaptive workload-management systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 1, pp. 221–231, Mar. 2011.
- [5] D. Diaz, A. Cesta, A. Oddi, R. Rasconi, and M.D. R-Moreno, "Efficient energy management for autonomous control in rover missions," *IEEE Computational Intelligence Magazine*, vol. 8, no. 4, pp. 12–24, Nov. 2013.
- [6] J. Ziomek, L. Tedesco, and T. Coughlin, "My car, my way: Why not? I paid for it!," *IEEE Consumer Electronics Magazine*, vol. 2, no. 3, pp. 25–29, Jul. 2013.
- [7] S. C. Huang, B. H. Chen, S. K. Chou, J. N. Hwang, and K. H. Lee. Smart car application platform [Online]. Available: <https://github.com/smartCarLab/smartCar>
- [8] H. Koschmieder, "Theorie der horizontalen sichtweite," in *Beiträge zur Physik der Freien Atmosphäre*. Munich, Germany: Keim & Nemnich, 1924.
- [9] S. C. Huang, B. H. Chen, and Y. J. Cheng, "An efficient visibility enhancement algorithm for road scenes captured by intelligent transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2321–2332, Oct. 2014.
- [10] S. C. Huang, J. H. Ye, and B. H. Chen, "An advanced single image visibility restoration algorithm for real-world hazy scenes," *IEEE Trans. on Industrial Electronics*, vol. 62, no. 5, pp. 2962–2972, May 2015.
- [11] J.-P. Tarel, N. Hautière, L. Caraffa, A. Cord, H. Halmaoui, and D. Gruyer, "Vision enhancement in homogeneous and heterogeneous fog," *IEEE Intelligent Transportation Systems Magazine*, vol. 4, no. 2, pp. 6–20, Summer 2012.
- [12] B. H. Chen, S. C. Huang, and J. H. Ye. "Hazy image restoration by bi-histogram modification," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 4, Article 50, Jul. 2015.
- [13] N. Hautière and D. Aubert, "Contrast restoration of foggy images through use of an onboard camera," in *Proc. IEEE Conf. Intelligent Transportation Systems*, pp. 601–606, Sept. 2005.
- [14] S. C. Huang, B. H. Chen, and W. J. Wang, "Visibility restoration of single hazy images captured in real-world weather conditions," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 10, pp. 1814–1824, Oct. 2014.
- [15] L. Wang, L. Xiao, H. Liu, and Z. Wei, "Variational Bayesian method for retinex," *IEEE Trans. on Image Processing*, vol. 23, no. 8, pp. 3381–3396, Aug. 2014.
- [16] K. S. Sim, C. P. Tso, and Y. Y. Tan, "Recursive sub-image histogram equalization applied to gray scale images," *Pattern Recognition Letters*, vol. 28, no. 10, pp. 1209–1221, Jul. 2007.
- [17] S. C. Huang and C. H. Yeh, "Image contrast enhancement for preserving mean brightness without losing image features," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 5-6, pp. 1487–1492, May-June 2013.
- [18] Z. Y. Chen, B. R. Abidi, D. L. Page, and M. A. Abidi, "Gray-level grouping (GLG): An automatic method for optimized image contrast enhancement-part I: the basic method," *IEEE Trans. on Image Processing*, vol. 15, no. 8, pp. 2290–2302, Aug. 2006.
- [19] T. Celik, "Spatial entropy-based global and local image contrast enhancement," *IEEE Trans. on Image Processing*, vol. 23, no. 12, pp. 5298–5308, Dec. 2014.
- [20] D. J. Jobson, Z. Rahman, and G. A. Woodell, "Properties and performance of the center/surround retinex," *IEEE Trans. Image Process.*, vol. 6, no. 3, pp. 451–462, Mar. 1997.
- [21] T. Celik and T. Tjahjadi, "Automatic image equalization and contrast enhancement using Gaussian mixture modeling," *IEEE Trans. Image Process.*, vol. 21, no. 1, pp. 145–156, Jan. 2012.
- [22] Y. T. Kim, "Contrast enhancement using brightness preserving bi-histogram equalization," *IEEE Trans. Consum. Electron.*, vol. 43, no. 1, pp. 1–8, Feb. 1997.

- [23] Y. Wang, Q. Chen, and B. Zhang, "Image enhancement based on equal area dualistic sub-image histogram equalization method," *IEEE Trans. Consum. Electron.*, vol. 45, no. 1, pp. 68–75, Feb. 1999.
- [24] F. C. Cheng, and S. C. Huang, "Efficient histogram modification using bilateral bezier curve for the contrast enhancement," *IEEE/OSA J. Disp. Technol.*, vol. 9, no. 1, pp. 44–50, Jan. 2013.
- [25] S. C. Huang, F. C. Cheng, Y. S. Chiu, "Efficient contrast enhancement using adaptive gamma correction with weighting distribution," *IEEE Trans. Image Process.*, vol. 22, pp. 1032–1041, Mar. 2013.
- [26] D. Gerònimo, A. M. Lòpez, A. D. Sappa and T. Graf, "Survey of pedestrian detection for advanced driver assistance systems," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1239–1258, Jul. 2010.
- [27] D. M. Gavrila and S. Munder, "Multi-cue pedestrian detection and tracking from a moving vehicle," *Int'l J. Computer Vision*, vol. 73, no. 1, pp. 41–59, 2007.
- [28] M. Bertozzi, A. Broggi, A. Fascioli, A. Tibaldi, R. Chapuis, and F. Chausse, "Pedestrian localization and tracking system with Kalman filtering," *Proc. IEEE Intelligent Vehicles Symp.*, pp. 584–589, 2004.
- [29] V. Philomin, R. Duraiswami, and L. Davis, "Pedestrian tracking from a moving vehicle," *Proc. IEEE Intelligent Vehicles Symp.*, pp. 350–355, 2000.
- [30] R. Arndt, R. Schweiger, W. Ritter, D. Paulus, and O. Lohlein, "Detection and tracking of multiple pedestrians in automotive applications," *Proc. IEEE Intelligent Vehicles Symp.*, pp. 13–18, 2007.
- [31] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564–577, May 2003.
- [32] C. T. Chu, J. N. Hwang, H. I. Pai, and K. M. Lan, "Tracking human under occlusion based on adaptive multiple kernels with projected gradients," *IEEE Trans. Multimedia*, vol. 5, no. 7, pp. 1602–1615, Nov. 2013.
- [33] I. J. Cox, "A review of statistical data association techniques for motion correspondence," *Int'l. J. Computer Vision*, vol. 10, no. 1, pp. 53–66, 1993.
- [34] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Automatic Control*, vol. 24, no. 6, pp. 843–854, Dec. 1979.
- [35] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE J. Oceanic Eng.*, vol. 8, no. 3, pp. 173–184, Jul. 1983.
- [36] A. Ess, B. Leibe, K. Schindler, and L. Van Gool, "Robust multiperson tracking from a mobile platform," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, no. 10, pp. 1831–1846, Oct. 2009.
- [37] M. Andriluka, S. Roth, and B. Schiele, "People-tracking-by-detection and people-detection-by-tracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Jun. 2008.
- [38] B. Leibe, K. Schindler, N. Cornelis, and L. VanGool, "Coupled object detection and tracking from static cameras and moving vehicles," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1683–1698, Oct. 2008.
- [39] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Jun. 2008.
- [40] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes, "Globally-optimal greedy algorithms for tracking a variable number of objects," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Jun. 2011.
- [41] Z. Wu, A. Thangali, S. Sclaroff, and M. Betke, "Coupling detection and data association for multiple object tracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1948–1955, Jun. 2012.
- [42] A. A. Butt and R. T. Collins, "Multi-target tracking by Lagrangian relaxation to min-cost network flow," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Jun. 2013.
- [43] A. Kundu, K. M. Krishna, and C. V. Jawahar, "Realtime multibody visual slam with a smoothly moving monocular camera," *Proc. IEEE Conf. Computer Vision*, pp. 2080–2087, Nov. 2011.
- [44] S. Song and M. Chandraker, "Robust scale estimation in real-time monocular SFM for autonomous driving," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Jun. 2014.
- [45] K. H. Lee, J. N. Hwang, G. Okapal, and J. Pitton, "Driving recorder based on-road pedestrian tracking using visual SLAM and constrained multiple-kernel," *Proc. IEEE Conf. Intelligent Transportation Systems*, pp. 2629–2635, Oct. 2014.
- [46] C. T. Chu and J. N. Hwang, "Fully unsupervised learning of camera link models for tracking humans across non-overlapping cameras," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 6, pp. 979–994, Jun. 2014.
- [47] K. H. Lee, J. N. Hwang, "On-road pedestrian tracking across multiple driving recorders," *IEEE Trans. on Multimedia*, vol. 17, no. 9, Sep. 2015.