Generating Arbitrary View of Vehicles for Human-assisted Automated Vehicle Recognition in Intelligent CCTV Systems

Youri Ku, Insu Kim and Kin Choong Yow
GIST College, Gwangju Institute of Science and Technology
Gwangju, Republic of Korea
e-mail: kyr2234@gist.ac.kr, ahinsutime@gist.ac.kr, kcyow@gist.ac.kr

***Abstract*—Intelligent closed-circuit televisions (CCTV) are CCTV systems that can perform Video Content Analysis (VCA). However, in the area of Automatic Vehicle Recognition, there is still no good algorithm to recognize a car based on its description. In this paper, we propose a novel algorithm that will take an image (or several) of a car, extract special markings (if any) from it, and then texture map it to a 3D model of the same car. With the texture mapped 3D model, we can rotate the car to any arbitrary view point, especially to the view of another CCTV, so that non-sophisticated image matching algorithms can match the image to the actual CCTV feed. We performed experiments on three cars with different body markings and the results show that we can achieve quite realistic images of the car at any arbitrary viewpoint. This system will have significant impact on the use of intelligent CCTVs.**

Keywords - 3D model reconstruct; drone; generate unseen view of object; image warping; texture map UVW; arbitrary view .

I. INTRODUCTION

CCTV cameras, or closed circuit television cameras, are undoubtedly one of the most pervasive devices used in security systems all over the world today. In fact, over the years, these devices did not just help the authorities to pursue criminals, they were also used to view and monitor traffic incidents, estimate crowd density, and even detect suspicious activities within private businesses and residences. It can even be mounted on an Unmanned Aerial Vehicle (UAV), e.g., the Aeryon Scout [1] and be instructed to fly to a different incident location.

It is an ill-informed opinion that all CCTV does is to provide the ability to review an event after it has already taken place or to ‘spy’ on passers-by. Modern technologies are able to analyze the video data captured, alongside with the use of triggers, to prompt security actions for certain events or situations. The term “Intelligent CCTV” is now loosely used to describe systems that have such Video Content Analysis (VCA) ability.

In a study by Ju and Yi [2], the global video surveillance market is a huge market picking up 9 billion dollars in 2010, and it is expected to achieve 11.3 billion in 2012 and 14.4 billion in 2015. Among them, the intelligent CCTV market aggregated 0.2 billion dollars in 2010, and projected to hit 0.3 billion in 2012 and 0.6 billion in 2015.

The intelligent CCTV is touted to change the way we interact or react to people. The Intelligent CCTV has several advanced capabilities, some of which includes:

- Human Face Recognition
- Car License Plate Recognition
- Point of Sale (POS) / Shrinkage detection
- Object Tracking
- Unattended Object Detection
- Traffic Monitoring, and
- Behavior Recognition

We observe that there is a severe lack of technologies in the area of Automatic Vehicle Recognition in Intelligent CCTV systems. The only technology that is used here seems to be the Car Licensed Plate recognition technology. While it is true that the license plate may be the only unique “feature” to identify a car, there may be many situations where it is not possible to obtain a clear view of the license plate of a car that is leaving/entering a town/city, or a car that is simply parked in a crowded car park.

Suppose you have a suspicious car that you want to track. How would you convey to the Police the information about the car? Apart from the license plate information, you will also give a description of the car (e.g., “A White Audi Q7 with stripes on the hood”, etc.). The Police will search the neighborhood first for a white Audi Q7, and after finding one, they will proceed to check if the car has stripes on its hood and whether the license plate matches.

This would be exactly the same for an Intelligent CCTV system. If we want the Intelligent CCTV system to automatically search an area for a car that matches our description, we need to provide it with more information than just the license plate number.

However, to communicate with the computer that you want a car “with stripes on the hood” is a notoriously difficult thing to do. The best option is to have an image of the car so that the computer can perform image matching on the CCTV feed. Another problem is that existing image matching algorithms do not perform well if the view of the object differs too greatly between the two images, or the background is drastically different.

This leads us to the motivation for our proposed solution. We want to develop an Intelligent CCTV system that takes at least one image of the vehicle that we want to find (additional images may provide views of the vehicle not seen in the first image), and with a very small amount of help from a human operator (to identify 5 to 9 points on the image), the system will be able to generate an arbitrary view of the car from a pre-defined 3D model that matches the view and background of the CCTV camera. With this new view, a non-sophisticated image matching algorithm will be able to find a successful match of the car with low false positive rates.

The rest of the paper is organized as follows: Section II discusses some related work, and Section III describes the proposed algorithm. Section IV shows our reconstruction results and Section V concludes the paper.

II. RELATED WORK

Intelligent CCTV surveillance systems make use of a variety of image and video processing technologies to exact the information they need for their tasks. Foresti et al. [3] identify moving vehicles in video data streams by subtracting the current frame from an estimate of the background scene, based on the idea that anything 'new' in the current frame must be the mobile vehicle(s). To cope with occlusion, they made use of a Kalman filter to provide better estimates of the place and time of the vehicle's emergence from the occluded zone. Greenhill et al. [4] proposed a method which significantly improved the accuracy of object tracking by utilizing knowledge about the monitored scene. Such scene knowledge includes the homography between the camera and ground planes and the occlusion landscape identifying the depth map associated with the static occlusions in the scene.

Lotufo et al. [5] proposed the ANPR (Automatic Number Plate Recognition) system that is based on Computer Vision. The system performs a detailed matching process between the extracted character features and the reference features (contained in a database) using a statistical nearest neighbor classifier. Saravi and Edirisinghe [6] present an approach to Vehicle Make & Model Recognition (VMMR) in CCTV video footage that uses CPD (coherent Point Drift) to remove skew of vehicles detected. They also proposed a LESH (Local Energy Shape Histogram) feature based approach for vehicle make and model recognition that uses temporal processing to improve reliability.

III. PROPOSED ALGORITHM AND IMPLEMENTATION

A. Problem Definition

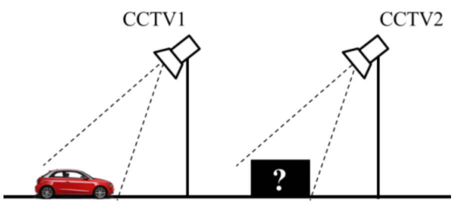


Figure 1. Problem Definition

Assume that we are living in a city that is equipped with Intelligent CCTV systems. In Figure 1, CCTV1 and CCTV2 are two arbitrary cameras located in the city. Each CCTV camera knows its own location and elevation as well as the inclination of the camera. A car is moving on the road and CCTV1 is tracking the car. However, since CCTV1 cannot move, the car will soon disappear from the field of view of CCTV1. Luckily, along the same road at some distance away there is another camera CCTV2, but the car is not yet in the field of view of CCTV2. Then how can CCTV1 communicate the car information to CCTV2 so that when the car comes into CCTV2's field of view, CCTV2 can continue to track it?

We identify that CCTV1 must provide at least one image of the car (with no restriction to its viewpoint) to CCTV2. Since CCTV2 knows its own position, elevation and camera inclination, it must generate a view of the car that matches its viewpoint of the road when the car begins to come into its field of view. Figures 2 and 3 illustrate two scenarios where the source image(s) can be obtained.

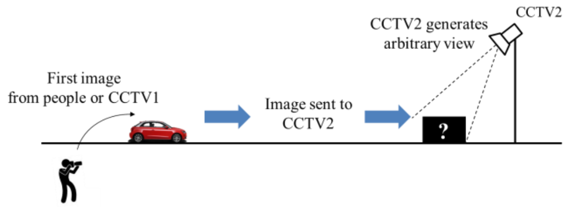


Figure 2. System with only one source image

In the first scenario, we have only one image of the car. The image may be generated by a person with a digital camera or mobile phone camera, or even by another intelligent CCTV camera (CCTV1). This image is then sent to CCTV2 where it will generate a new view of the car that matches its own viewpoint of the road.

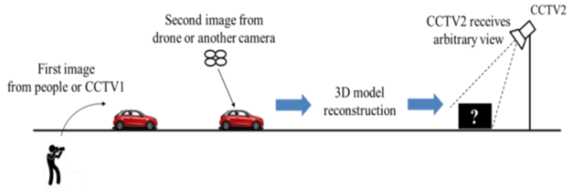


Figure 3. System with two or more source images

In the second scenario, we have two or more images. The first image may be generated by a person with a digital camera or mobile phone camera, or by another intelligent CCTV camera (CCTV1). The second image may be taken by a drone (i.e., UAV) or by another CCTV camera. If the drone is the source of the second image, it can offer not only the top view but also many views in different angles because it is a flying camera. These images are then sent to CCTV2 where it will generate a new view of the car that matches its own viewpoint of the road.

B. System overview

In this section, we describe our proposed system. We assume that the car to be tracked by CCTV2 has some special markings to differentiate it from another car of the same type (e.g., see Figure 5). If no special markings are available, then there may be many cars that look exactly the same and it may not be possible to correctly identify the right car to track. In addition, the intelligent CCTV camera (CCTV2) needs to receive a small amount of help from a human user (to input 5 – 9 corresponding points for texture mapping) as current Computer Vision correspondence algorithms are not robust enough yet to perform this task automatically. The overview of the system is shown in Figure 4.

 ISBN: 978-1-61208-418-3

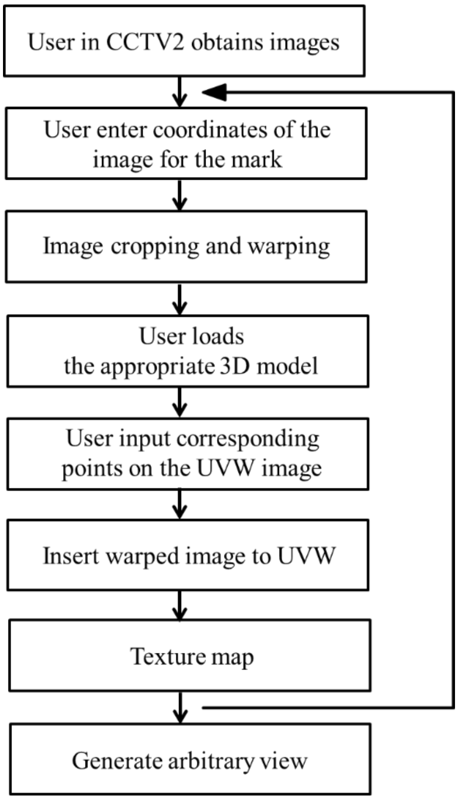


Figure 4. System overview

C. *Special marking extraction*

First, when the human user in CCTV2 receives the image(s), he needs to identify what make and model of the car it is. Although there are existing VMMR algorithms (e.g., Saravi and Edirisinghe [6]) for vehicle make and model recognition, they are not robust enough yet to perform this task automatically. Then, the human user will determine if there are any special markings on the car and on which panel they are (e.g., hood). If special markings exist, the human user will determine the coordinates of a 4-point polygon (representing a skewed rectangle) that covers the markings, and enters them into the system. The system will then automatically crop and warp the images into a rectangle.

Figure 5 shows an Audi Q7 that has stripes running down the hood. The four points in red (on the hood near to the stripes) in Figure 5 are the four points that the human user needs to enter into the system.

Figure 5. Audi Q7 with strips on the hood

Figure 6 shows an Opel Corsa that has yellow flame patterns on a blue plate. In this case, two images were available for the same car. If only one image was available (e.g., Figure 6(a)), then we could only extract the hood and driver-side door patterns and we will not know that there are also patterns on the trunk and the passenger-side door. This may lead to an incorrect 3D model reconstruction. To increase our chances for obtaining a correct 3D model, we can make certain assumptions (e.g., that the flame mark exists on the doors on both side of the car) and generate several candidate models for recognition.

(a) (b)

Figure 6. Opel Corsa with Flame patterns

If both images (Figure 6(a) and (b)) are available then we can completely reconstruct the 3D model of the Opel Corsa. The white circles in Figure 6 are the points of the polygon where the user has to identify and enter into the system.

(a) (b)

Figure 7. Opel Corsa with Green Stripes

Figure 7 shows another Opel Corsa that has a different pattern from Figure 6. This Opel Corsa has a black plate with a green stripe mark running from the hood to the trunk. In this case, there are also two images of the same car. The second image (Figure 7(b)) represents an image taken by a drone or from a CCTV camera that is mounted at the top of a building. If this image was not available, we will face the same problem as the Opel Corsa with Flame patterns. The white points in Figure 7(a) and (b) are the points of the polygon where the user has to identify and enter into the system.

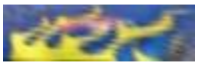
D. *Image warping*

After the human user has identified the coordinates of the special markings, the system will perform image warping automatically using the OpenCV *warpPerspective* function and transform the polygon in the image into a rectangle. This is necessary for the next stage of texture mapping for the 3D model. Figures 8, 9 and 10 show the warped image of the pattern for the three cars in our experiments.


(a) (b)

Figure 8. Audi Q7 with warped image of pattern

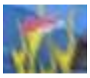
Figure 8(a) is original car image of Audi Q7 which is not modified at all. The Figure 8(b) is the warped image of extracted black stripe mark from Figure 8(a).

(b)

(a) (c)


(e)



(d) (f)

Figure 9. Opel Corsa with warped image of flame pattern

Figures 9(a) and 9(d) are original car image of Opel Corsa with flame pattern. Figure 9(b) is the warped image of extracted flame mark on the front hood of the car from Figure 9(a). The Figure 9(c) is warped image of flame mark on the left door in Figure 9(a). The Figure 9(e) is warped image of mark extracted from right door of car in Figure 9(d), and Figure 9(f) is warped image of mark on car trunk in Figure 9(d).


(a) (b)



(c) (d) (e)

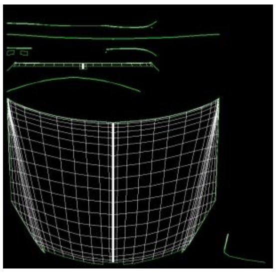
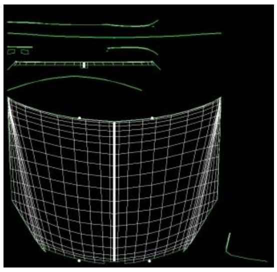
Figure 10. Opel Corsa with warped image of green stripes

Figures 10(a) and 10(c) are the raw images of car. Figure 10(b) shows the warped image of extracted mark which was on the front hood in Figure 10(a) originally. In the same way Figures 10(d) and 10(e) are warped image of extracted mark from Figure 10(c). Figure 10(d) is the mark which was on the top side of the car in Figure 10(c), and Figure 10(e) is the mark on the car trunk.

E. Texture Mapping

The next step is to apply the warped pattern images to the 3D model of the car (texture mapping). There are numerous websites that offer 3D car models for download (some are freeware and some are payware) and they come in a variety of formats (3DSmax, Maya, Wavefront, etc.). We obtain the 3D models for our three experiments from a website [7].

After we have downloaded the corresponding car model, we need to perform a UVW unwrap operation (from any 3D modeling software such as 3DSmax) to obtain the UVW map of the model. The UVW map allows us to perform texture mapping onto the 3D car model. An example of the UVW map of the Audi Q7 is showed in Figure 11(a).

(a) (b)

Figure 11. UVW map of the Audi Q7

To perform the texture mapping, we need to attach the warped image to the UVW map of the car. Here, the human user has to enter another 4 points on the UVW map to show where the warped image will go. Figure 11(b) shows the 4 points that the human user have entered.

After entering the 4 points, the warped image will be automatically resized and then "pasted" onto the UVW map. In addition, the human user needs to specify one more point in the image (that has the color of the body of the car) so that all the panels in the UVW can be filled with this color. This completes the texture mapping process on the UVW map and we are ready to reconstruct the 3D car. Figures 12, 13 and 14 show the completed UVW maps of all the three cars in our experiment.

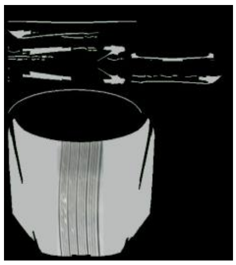
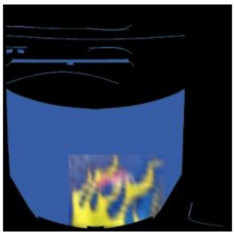
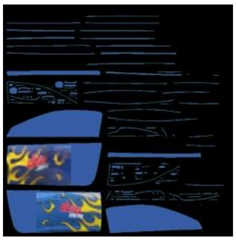
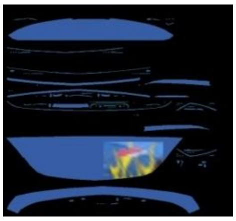


Figure 12. Completed UVW map of the Audi Q7

In Figure 12, the UVW map of the Audi Q7 is completed by pasting Figure 8(b) on the chosen part of raw UVW map which is Figure 11(b). Since the size of warped mark and chosen part of UVW map are different each other, the algorithm performs resizing process before pasting.

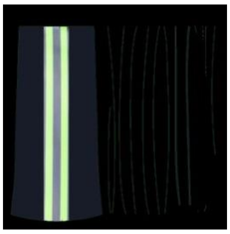
(a) (b)



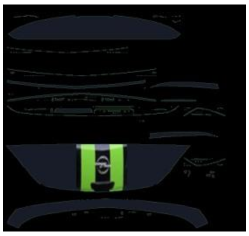
(c)

Figure 13. Completed UVW map of the Opel Corsa with Flame pattern

Figures 13(a), 13(b) and 13(c) are completed UVW map of Opel Corsa with flame pattern. Each of them is generated by attaching warped images of part D to the region of UVW which is chosen by user. Figure 9(b) is used for Figure 14(a), Figures 9(c) and 9(e) are used for Figure 14(b), and Figure 9(f) is used for Figure 13(c).

(a) (b)



(c)

Figure 14. Completed UVW map of the Opel Corsa with Green Stripes

Figures 14(a), 14(b) and 14(c) are the completed UVW map of the Opel Corsa with green stripes. The resized Figures 10(b), 10(d), and 10(e) are attached to corresponding position of UVW map of Opel Corsa.

F. *Reconstruction of the 3D model in the Required View*

We assume that the CCTV camera (CCTV2) position, elevation and inclination are already known. Let the CCTV camera position be (x, y, z). In addition, we also assume that the position of car in CCTV2's field of view is also known to be (x_c, y_c, z_c). Figure 15 shows the relationship between the two points.

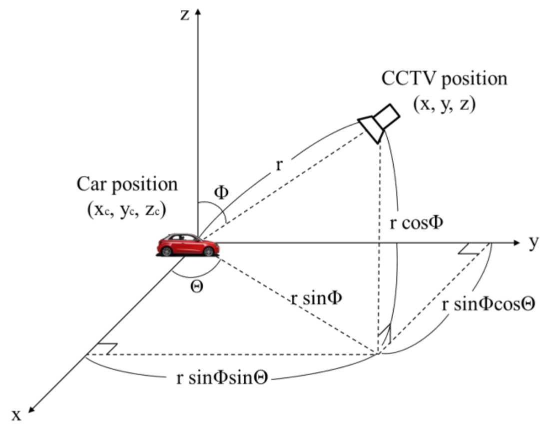


Figure 15. Construction of arbitrary views

With the information on the car position and the camera position, we can generate the expected view of the car from CCTV2 using the *gluLookat* function in the OpenGL library. By rotating the 3D reconstructed model, we can generate many arbitrary views of the car at different angles.

The final step is to merge the expected view of the car to the background image seen by CCTV2. This is a very simple process since the generated view of the car has a black background, so all we need to do is to replace the background image pixels by the non-zero car image pixels at the desired location.

IV. EXPERIMENTAL RESULTS

A. *Result of reconstucting 3d model*

Figure 16 shows the results of reconstructing the 3D model of the three cars to the same view as the original images so that we can verify that the result is correct.


(a) (b)


(c) (d)


(e) (f)

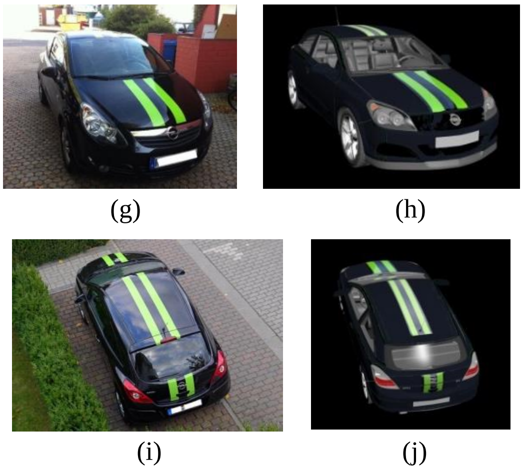


Figure 16. Reconstruction results

Figures 16(a), 16(c), 16(e), 16(g) and 16(i) are the raw car images of Audi Q7 and Opel Corsa which are same with Figures 8(a), 9(d), 9(a), 10(a) and 10(c) each. Figures 16(b), 16(d), 16(f), 16(h) and 16(j) are the reconstructed 3D model of each car.

B. Result of generating new view

Figure 17 shows the results of generating new arbitrary views of the three cars.

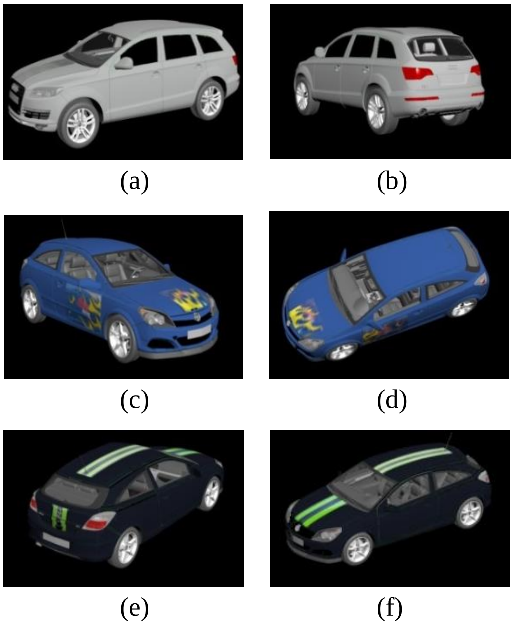


Figure 17. New arbitrary views of the cars

Figures 17(a) and 17(b) are the arbitrary view of Audi Q7. Figures 17(c), 17(d), 17(e) and 17(f) are different view of Opel Corsa with flame pattern and green stripe. We can see that all the three cars in our experiment look realistic.

C. Merging with the CCTV background image

Figure 18 shows the result of merging the generated car image to the CCTV background image. The Audi Q7 is placed on the side of a road in Figure 18(a). Figures 18(b) and 18(c) show the Opel Corsa with flame patterns is placed at the top left corner of a junction, and the Opel Corsa Green Stripe is placed at the top right hand corner of the junction. Figure 18(d) is the zoom-in image of yellow box in Figure 18(c).

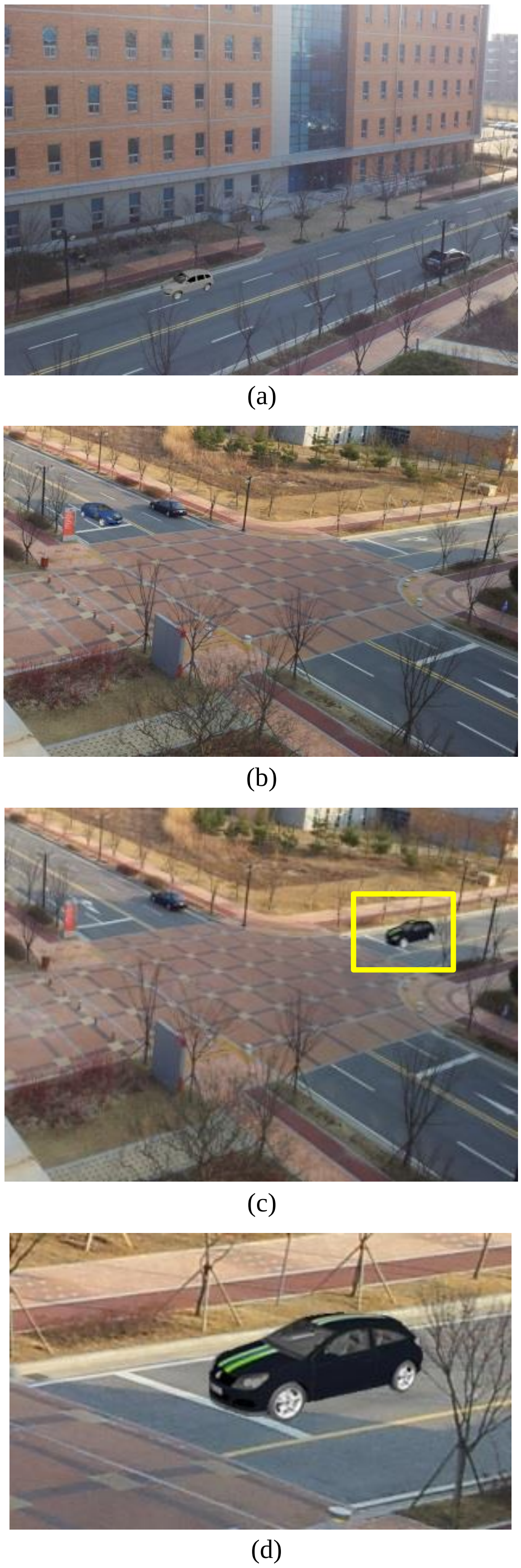


Figure 18. Merging with CCTV background image

We can see that the car images blend in very well to the CCTV background image and the resulting image looks quite realistic. We also showed a zoomed-in image of the Opel Corsa with green stripes to show that the

 ISBN: 978-1-61208-418-3

results look equally good when zoomed in.

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed a new algorithm for generating an arbitrary view of a car from at least one image, taken from separate digital camera, mobile phone camera, or CCTV camera. Our system requires a little help from a human user (to identify the car's make and model, as well as to select and input 5 – 9 corresponding points for texture mapping). After the texture mapping is completed, we can generate any arbitrary view of the car, most importantly the view from the intelligent CCTV camera that will be tracking the car.

Although the resulting image looks realistic, this can be improved further by considering the position of the sun as well as the time of the day. At different times of the day, the position of the sun will cause different reflections as well as shadows on the car, causing the image to look different from what it was. This will be the focus of our future work in this project.

REFERENCES

[1] Aeryon Labs Inc. [online], available at http://www.aeryon.com/products/avs/aeryon-scout.html. Retrieved Mar 21, 2015.

[2] Y. W. Ju and S. J. Yi, " Implementing Database Methods for Increasing the Performance of Intelligent CCTV", International Journal of Security and Its Applications Vol.7, No.5 (2013), pp.113-120. http://dx.doi.org/10.14257/ijsia.2013.7.5.09.

[3] G. Foresti, C. Micheloni, L. Snidaro, P. Ramagnino, and T. Ellis, "Active Video-based Surveillance System," IEEE Signal Processing Magazine, March 2005, pp. 25-37.

[4] D. Greenhill, J. Renno, J. Orwell, and G. A. Jones, "Learning the Semantic Landscape: Embedding scene knowledge in object tracking," Realtime Imaging, Special Issue on Video Object Processing, Volume 11, Issue 3, June 2005, Pages 186–203. doi:10.1016/j.rti.2004.12.002

[5] R. A. Lotufo, A. D. Morgan, and A. S. Johnson, "Automatic number-plate recognition", IEE Colloquium on Image Analysis for Transport Applications, Feb 1990, pp.1-6.

[6] S. Savari and E. A. Edirisinghe, "Vehicle Make and Model Recognition in CCTV footage", 18th International Conference on Digital Signal Processing (DSP), 2013, July 2013, pp. 1-6.

[7] Crazy 3D Free.com, Retrieved from http://www.crazy3dfree.com on 2015.6.7.