

A Real-Time Vehicular Vision System to Seamlessly See-Through Cars

Francois Rameau^(✉), Hyowon Ha, Kyungdon Joo, Jinsoo Choi,
and InSo Kweon

Electrical Engineering Department, KAIST, Daejeon, South Korea
{frameau,hwha,kdjoo,jschoi,iskweon}@rcv.kaist.ac.kr

Abstract. Overtaking accidents typically occur when the rear car intends to overtake the front car with limited visibility. This lack of visual information is often attributed to the occlusion caused by the front vehicle. Indeed, in many situations the front car hides the presence of obstacles, such as pedestrians or other cars. Nowadays, the generalization of digital camera embedded automotives represents a great potential to reduce the number of these deadly accidents. Thus, we propose a novel collaborative cars method which allows a driver to literally see through the front vehicle to assist in overtaking manoeuvres. In the studied scenario, both cars are equipped with cameras (stereo and monocular cameras for the front and the rear cars, respectively) and share data through an appropriated wireless communication system. Our method generates a seamless transparency effect from the rear car viewpoint using tri-focal tensor image synthesis where the poses of the cameras are estimated using a marker-based pose estimation. In this article, we present an efficient framework designed to reduce the quantity of information to be transferred between the vehicles and to achieve real-time performances (15 fps). Furthermore, our system is assessed through multiple experiments in controlled and real conditions.

Keywords: Augmented reality · Image synthesis · See-through · Marker-based approach · Real-time

1 Introduction

Nearly 1.3 million people die in road crashes each year, while 20 to 50 million people are injured or disabled. According to [1], overtaking contributes to about 1% of all the road accidents but represents a very high percentage of deadly accidents due to frontal impact and high speed - which are typical of overtaking situations. These accidents have multiple causes which are enumerated in [2], where the authors identified the lack of visibility (poor observation) as the major factor in overtaking collisions. These accidents are even more common in low- and middle-income countries due to poor road infrastructures and outdated vehicles. Furthermore, most of these dangerous situations involve large vehicles such as trucks.

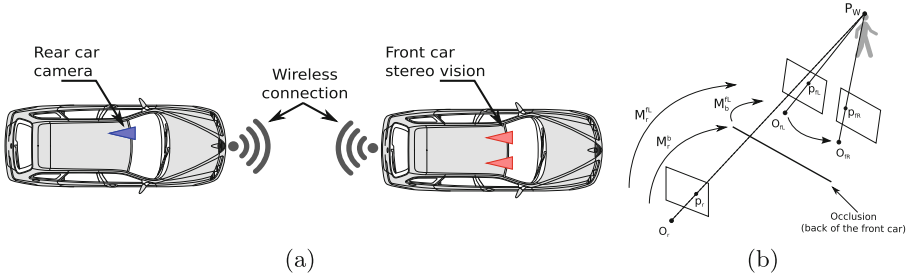


Fig. 1. (a) Main scenario. (b) Model of the problem.

This paper concerns the collaboration between two vehicles equipped with cameras and connected together via a wireless network such that data can be transferred. In real case scenarios, these cars are following each other. As a consequence, the vision of the rear car is occluded by the front car itself, making it impossible to see any hypothetical obstacles - such as pedestrians - (see Fig. 1(a)). Ultimately, the purpose of this work is to overcome this occlusion, in other terms, to see through the front car in order to prevent eventual accidents. Despite the important impact of such technology for automotive safety, this application did not attract much attention yet.

Lately, overtaking accidents have received a renewed interest through the actions of large companies such as Samsung, which installed large LCD screens on the back of their trucks to display the front view of the vehicle¹. This solution is very practical and efficient but has a very high cost that most of the road transportation companies cannot afford.

A cheaper option is to display the images acquired from the front vehicle directly inside the following cars through an appropriate hardware setup to transfer the data and to visualize it. Olaverri-Monreal *et al.* [3] studied the impact of such technology on drivers' behaviours through simulated experiments. According to their work, such type of system positively impacts the driver's reactions. They also proved the feasibility of the system in real conditions. Therefore, this system is very interesting to provide more information to the drivers thanks to a live video-streaming. Nonetheless, the change of view-point can be confusing to the driver and can cause some miss-interpretation of the scene. For that reason, this work focuses on a more natural way to display the on-coming obstacles occluded by the front vehicle. Indeed, we propose to see-through the front vehicle literally speaking by generating a synthetic non-occluded image using images from both vehicles. Such approaches have been investigated in the past, for instance, in [4, 5] the perspective projection is approximated using the distance between cars (obtained with markers) and the size of the front vehicle, however the obtained result is nothing but a rough approximation of the geometry while we aim to see through the vehicle with a seamless transparency effect. In [6], the authors fused a large number of sensors between multiple vehicles,

¹ <https://www.youtube.com/watch?v=ZetSRWchM4w>.

the cars' poses are determined using a laser rangefinder and the synthetic view is generated using ground plane matching, thus only a very small portion of the view is transferred which does not carry relevant information. Finally, the most closely related paper is [7], where the authors propose to use a rough depth map (obtained from a laser scanner on the front vehicle) and a car localization method (based on GPS and laser scanners) to generate a synthetic unoccluded view, while our approach only relies on cameras. Moreover, [7] assumes various hypothesis on the structure of the scene; indeed, the vehicles have to lie on the same plane to be localized efficiently and the approximated depth map generation does not perform well in complex environments. Whereas our approach provides a full transparency effect without geometrical hypothesis on the scene and thus allows a wider number of scenarios (slope, traffic jam, urban canyon, etc.). In fact, in this paper, we describe a versatile framework based on marker-based pose estimation and image synthesis using a tri-focal tensor to see through cars in real-time. It should also be noted that despite the use of markers, the entire method is very generic and the beacons placed on the front car can be replaced by the CAD of the vehicle.

This paper is organised as follows. Section 2 is dedicated to the problem statement and notations, in the Sect. 3 we explain in detail the different steps of our algorithm, while the Sect. 4 provides implementation information necessary to reach real time performances. Our results are available in Sect. 5. Finally this paper ends with a short conclusion.

2 Problem Statement

Given two cars following each other, our goal is to see through the front vehicle from the rear car viewpoint in real time. To this purpose, our system requires three cameras, two are embedded in the front car (to estimate the 3D structure of the scene) and one is mounted on the rear car. With such setup it is clear that the 3D reconstruction from the front car can be reprojected in the rear car image to create an unobstructed picture. This strategy has the great advantage to provide a seamless transparency effect. However, solving this problem is not straightforward.

Figure 1(b) is roughly depicting the scenario we face at time t . In this scheme, \mathbf{O}_r is the camera on the rear car, while \mathbf{O}_{fL} and \mathbf{O}_{fR} are respectively the left and right cameras of the stereo rig mounted on the front car. The occlusion illustrated by a thick line in the figure, is the occlusion generated by the front car. As mentioned previously, the re-projection of the occluded area on the rear camera image is highly dependent upon the 3D geometry of the scene. The front car stereo-vision system is able to provide this very desirable information by triangulating the corresponding image points \mathbf{p}_{fL} and \mathbf{p}_{fR} in order to obtain the 3D point \mathbf{P}_W . The problem lies in the re-projection of this 3D point on the rear car camera. If the poses between the front and the rear cameras are known (namely \mathbf{M}_r^{fL}), then it is possible to construct a tri-focal tensor. This tri-focal tensor can be further utilized to generate a synthetic image of the occluded

region by a point transfer from the front car to the rear car camera. This type of transfer is specific to tri-focal tensor and is called the point-point-point transfer. Note that a dense disparity map is needed in order to generate a proper synthetic image.

Therefore, our method consists of two main steps: the pose estimation of the cameras and the image synthesis. For the pose estimation of the cameras, we propose an appropriate technique to compute the transformation \mathbf{M}_r^{fL} . Since it is very challenging to match the images from the rear and the front car (mainly due to the large distance between the cars) we suggest to compute this transformation in two stages. Firstly, a Hand-Eye calibration process is providing the fixed rigid transformation \mathbf{M}_b^{fL} . Secondly, for every single time t the pose estimation of the front car \mathbf{M}_r^b is computed thanks to a marker-based approach. Thus, it is obvious that the transformation \mathbf{M}_r^{fL} is nothing but the composition of these two transformations:

$$\mathbf{M}_r^{fL} = \mathbf{M}_b^{fL} \mathbf{M}_r^b. \quad (1)$$

From this pose (\mathbf{M}_r^{fL}), the corresponding points, \mathbf{p}_{fL} and \mathbf{p}_{fR} , can be transferred in the rear camera image at the pixel coordinates \mathbf{p}_r , which, we can call the synthetic image point.

3 Methodology

In this section we present the two main steps of our method, the pose estimation of the cameras (including the calibration of our system) and the generation of a synthetic image using tri-focal tensor.

3.1 Pose Estimation of the Cameras

The previously mentioned point-point-point transfer is possible if and only if the inter-camera poses are known. In our situation it is difficult to estimate the tri-focal tensor using point correspondences between the 3 cameras (\mathbf{O}_{fL} , \mathbf{O}_{fR} and \mathbf{O}_r) -since the images acquired from the front and rear car can be very dissimilar.

Nonetheless, it is possible to determine the rigid transformation between the back of the front car and its cameras. Then, knowing the pose of the front car with respect to the rear car camera leads to the estimation of the poses of the entire system. In this section we briefly present our solutions to calibrate our system and to estimate the pose of the front car in real time.

Hand-Eye Calibration of the Front Car. A car intrinsically being a rigid structure, it is possible to calibrate the back of the front car with respect to its cameras. It means, solving the rigid transformation \mathbf{M}_b^{fL} . This type of problem is tackled under the name of Hand-Eye calibration, it is often solved as a homogeneous matrix equation of the type $\mathbf{AX} = \mathbf{XB}$ [8]. This particular formulation is very interesting since it allows to find the rigid transformation without using any

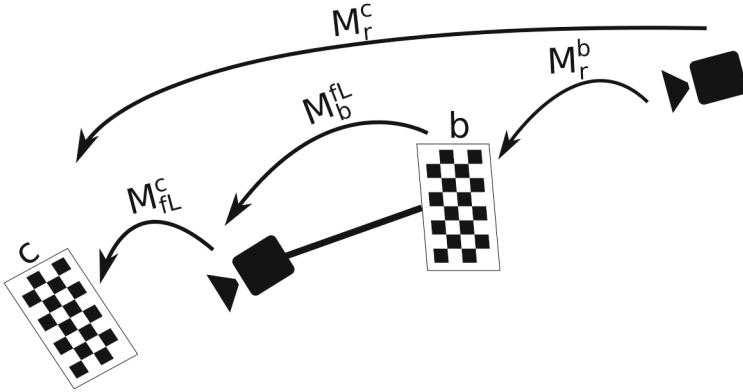


Fig. 2. Hand-Eye calibration scenario (only the left camera has been depicted for the sake of clarity)

inter-camera correspondences [9]. Unfortunately, this solution is very sensitive to degenerated motions. In our case the vehicle usually performs motions on a flat surface which makes it impossible to solve the vertical translation axis. Instead, we developed a more practical approach but where correspondences between the front and rear cameras are necessary.

For our calibration, two checkerboards are needed, one (called b in this paper) is fixed on the back of the front car while the second one c is located in the field of view of the front car camera (see Fig. 2). In case only one set of images is available, solving the desired transformation \mathbf{M}_b^{fL} is straightforward. In fact, as depicted in Fig. 2, the transformations \mathbf{M}_r^b and \mathbf{M}_{fL}^c can be directly computed thanks to the checkerboards. The remaining transformation \mathbf{M}_r^c can be estimated afterwards by removing the vehicle from the field of view of the rear camera. Finally the computation of \mathbf{M}_b^{fL} is found through the following matrix multiplication:

$$\mathbf{M}_b^{fL} = (\mathbf{M}_{fL}^c)^{-1} \mathbf{M}_r^c (\mathbf{M}_r^b)^{-1}. \quad (2)$$

The weakness of the described method lies in the large distance between the front checkerboard and the rear camera which can lead to an inaccurate corner detection. To increase the robustness of the approach, multiple sets of images have to be acquired in order to ensure a good estimation of \mathbf{M}_b^{fL} . Hence, a non-linear process minimizing the re-projection error over all the images is applied. In practice, the result from a single set of images is sufficient to initialize the refinement and to ensure a good convergence of the algorithm. In this work, the Levenberg-Marquardt algorithm is utilized to optimize the re-projection error.

Pose Estimation of the Front Car. We propose an efficient marker-based approach to determine, \mathbf{M}_r^b , the pose of the front car with respect to the rear car camera.

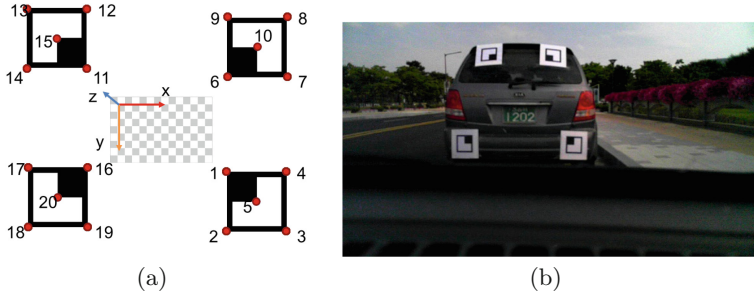


Fig. 3. (a) Four markers with 20 corner points used for pose estimation, (b) the markers mounted on the car

The approaches which solve this problem are commonly called “3D model-based pose estimation”. A large number of techniques taking advantage of various types of features have already been designed by the community. For instance using points [10], lines [11] or edges [12].

Most of the previously cited approaches are very accurate, however, a complete 3D model of the object of interest is needed in order to estimate its pose. In our case, it is difficult to know the full 3D model of the front car. Thus, we propose to use markers to estimate the pose between cars. The use of markers ensures an accurate, fast and robust localisation of the car, furthermore they are generic and can be setup on various type of vehicles. The markers employed for our pose estimation consist in a white square with thick black margins and containing a black square covering one of its quarters (see Fig. 3). Thus, five corners per marker can be detected and tracked; the four external corners and one in the center of the marker. To ensure a proper pose estimation, four markers are distributed on the back of the front car (see Fig. 3(b)), which represent a total of 20 points to compute the transformation \mathbf{M}_r^b . Each of these beacons can be efficiently detected using a basic square detection algorithm [13], while their ID is determined by estimating their orientations (0° , 90° , 180° and 270°). When one marker cannot be detected, its points (from the previous frame) are tracked using the KLT algorithm [14].

Note that in order to compute the pose of the set of markers (the back of the front car), the 3D position of their corners have to be expressed in the same referential coordinate system. In our case, we chose the checkerboard already utilized for the Hand-Eye calibration to be this reference (see Fig. 3). Thus, a camera captures both the checkerboard and the markers simultaneously from various viewpoints. For every single image, an accurate pose estimation of the camera is performed using the checkerboard. Therefore the 3D position of the corners can be accurately computed using a bundle adjustment based technique. After the calibration step, the checkerboard can be removed from the vehicle. With our experimental setup mounted on a car, 3000 images containing 20 corner points have been acquired. The computed mean reprojection error after optimization is 0.31 pixel.

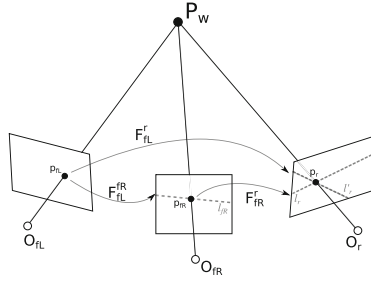


Fig. 4. Epipolar relationships among a tri-focal tensor

Finally, to estimate the transformation \mathbf{M}^b , we solve a Perspective-n-Point (PnP) problem using the EPnP algorithm described in [15]. Under certain conditions, false detections of the markers can affect the pose estimation. Thus, the robust pose estimation is ensured by a RANSAC process in order to efficiently remove the outlier points. Therefore, this robust estimation is refined by minimizing the re-projection error via a non-linear optimization scheme.

3.2 Image Synthesis Using Tri-Focal Tensor

The geometry of a triplet of images can be modelled using a tri-focal tensor $\mathbf{T}^{i,j,k}$ [16]. $\mathbf{T}^{i,j,k}$ defines the epipolar geometry between the three views indexed i, j, k in a $3 \times 3 \times 3$ cube matrix (see Fig. 4). It is the generalization of the fundamental matrix for a set of 3 views. From this formalism it is possible to transfer lines and points between views. In this section, we explain the computation of such kind of tensor from calibrated cameras, but also the point transfer by $\mathbf{T}^{i,j,k}$ and the image synthesis from a dense disparity map.

Tri-Focal Tensor from Projection Matrices. The trifocal tensor $\mathbf{T}^{fL,fR,r}$ linking the three perspective calibrated views fL , fR and r can be expressed as follow:

$$\mathbf{T}^{fL,fR,r} = [\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3], \quad (3)$$

$$\mathbf{T}_1 = \mathbf{M}_{fL}^{fR} [(^2\mathbf{M}_{fL}^{fL})^T \cdot ^3\mathbf{M}_{fL}^{fL} - ^3\mathbf{M}_{fL}^{fL} \cdot ^2\mathbf{M}_{fL}^{fL}]_{[\times]} (\mathbf{M}_{fL}^r)^T, \quad (4)$$

$$\mathbf{T}_2 = \mathbf{M}_{fL}^{fR} [(^3\mathbf{M}_{fL}^{fL})^T \cdot ^1\mathbf{M}_{fL}^{fL} - ^1\mathbf{M}_{fL}^{fL} \cdot ^3\mathbf{M}_{fL}^{fL}]_{[\times]} (\mathbf{M}_{fL}^r)^T, \quad (5)$$

$$\mathbf{T}_3 = \mathbf{M}_{fL}^{fR} [(^1\mathbf{M}_{fL}^{fL})^T \cdot ^2\mathbf{M}_{fL}^{fL} - ^2\mathbf{M}_{fL}^{fL} \cdot ^1\mathbf{M}_{fL}^{fL}]_{[\times]} (\mathbf{M}_{fL}^r)^T, \quad (6)$$

where ${}^i\mathbf{M}$ is the i^{th} row of the matrix \mathbf{M} and $\mathbf{X}_{[\times]}$ is the skew-symmetric matrix of the vector \mathbf{X} .

Point-Point-Point Transfer. As discussed before, if the tri-focal tensor is known and if a correspondence is found on at least two cameras, then it is

possible to re-project this point on the third view of the tensor. A formalism involving two fundamental matrices can be utilized in order to achieve this point transfer in the following manner:

$$\mathbf{p}_r = (\mathbf{F}_{fL}^r \mathbf{p}_{fL}) \times (\mathbf{F}_{fR}^r \mathbf{p}_{fR}), \quad (7)$$

with \mathbf{F}_{fL}^r and \mathbf{F}_{fR}^r respectively the fundamental matrices between the rear camera and the the left and right camera of the stereo-vision system. Nonetheless, this epipolar transfer is subject to many degenerated configurations as defined in [16].

The tri-focal tensor transfer can overcome these cases. Considering that the pose between the two views \mathbf{O}_{fL} and \mathbf{O}_{fR} is known (calibrated stereo-vision system). The point \mathbf{p}_{fR} in the second camera can be expressed with its epipolar line \mathbf{l}_{fR} , then the transfer can be simply expressed under the following point-line-point transfer: $\mathbf{p}_r^k = \mathbf{p}_{fL}^i \mathbf{l}_{fR}^j \mathbf{T}^{j,l,k}$. If this expression is used as it is, then, similar degenerated cases occurring with the epipolar transfer are faced. The easiest and more robust way to overcome this, is to use the orthogonal line of \mathbf{l}_{fR} passing by the point \mathbf{p}_{fR} . A detailed explanation is provided in [16].

Image Synthesis from Disparity Map. All the pixels composing the images from the stereo rig, can be transferred in order to create a synthetic image from another viewpoint (*i.e.* from the rear car viewpoint). As explained previously, this synthetic image can be obtained only if a dense matching between the two images from the stereo rig is available. Many works have focused on this pixel-to-pixel matching using stereo-vision systems, nonetheless a very few of them allows real time performances. In this paper we use [17], which has the advantage to provide a quasi-dense disparity map in real-time. However, the disparity values for poorly textured surfaces and far away elements are difficult to be estimated. In such scenarios, the very distant objects of the scene (lying at the horizon) as well as the sky, often have no disparity value associated with. This leads to a poor synthetic image quality which is difficultly readable for a human being.

To overcome this problem we propose a simple solution which consists of forcing a very small value to the undetermined disparities. This results in a more realistic synthetic image which is more appropriate for a good scene understanding. Other post-processing algorithms are also applied, such as a simple interpolation of the synthetic image to fulfil the texture-less part of the image and to deal with stereo shadow.

4 Implementation Details

To reach real time performances we designed our algorithm to share the computational cost between both computers embedded in the rear and the front car. The rear car processes are fully dedicated to markers detection, pose estimation and image stitching, while the front car generate the synthetic image accordingly with the information received from the rear car.

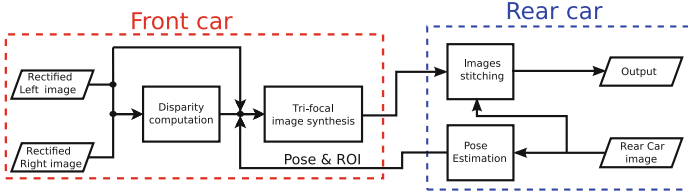


Fig. 5. Algorithm

The particular design of our algorithm also allows to optimize the quantity of data to be transferred between the cars. Indeed, we reached an optimal solution which drastically speed up the data transfer by only transmitting the relevant information. The rear car is simply sending the relative pose between cars and the localization of the ROI (the occlusion) in the image to the front car. These data are analysed to generate a small synthetic image (only covering the occluded region), thereafter this patch is compressed (using jpeg compression) and transferred back to the rear car to be stitched to the current image.

In case of wireless data transfer between moving cars, the distance between the communication platforms reduces the bandwidth, this effect is directly compensated by our solution since the size of the patch is also decreasing with respect to the distance between cars (the apparent size of the front car is shrinking with distance). The Fig. 5 depicts the whole algorithm. This algorithm has been fully developed in C++ and integrated in the middleware Robot Operating System (ROS).

5 Results

In this section, we present a series of experiments with real data emphasizing the efficiency of our method through quantitative and qualitative results. For our experiments, we utilize two cars. The front one is equipped with a *Bumblebee 2* (manufactured by *PointGrey*) in order to acquire well-synchronized stereo images. This sensor provides two 1024×768 p color images (downscaled to 512×384 p for speed issue) at 20 frames per second. A low cost USB color camera (Microsoft Lifecam Cinema) with a spatial resolution of 640×360 p is mounted on the rear vehicle. Since our algorithm does not need high bandwidth network, a standard Wi-Fi 802.11n network is used for our tests in outdoor condition; while most of the multi-vehicle collaboration applications require the use of an expensive Wi-Fi 802.11p network.

5.1 Assessment of Our System with Real Images

To evaluate the accuracy of our method with real images, we propose to estimate the transfer error of the synthetic points through an experimental setup. After the calibration of our system, a checkerboard is installed in front of the car

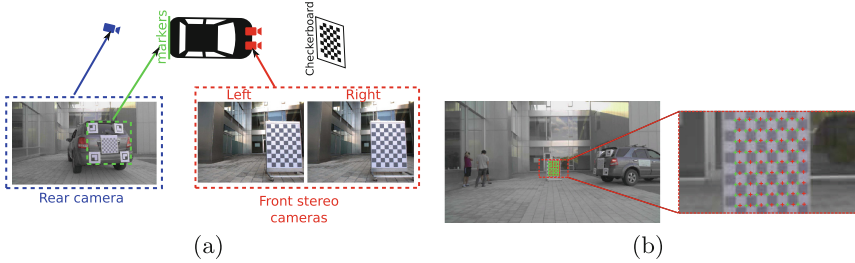


Fig. 6. (a) Experimental setup to test the accuracy of our point transfer; (b) Re-projection result for the second set of images, the green and red crosses are respectively the ground truth and the transferred points. (Color figure online)

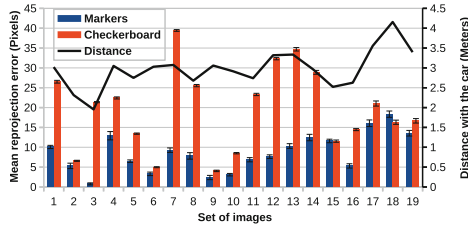


Fig. 7. Mean and standard deviation of the re-projection error of our system for 19 sets of images. The distance between the markers and the rear camera is also available in green (Color figure online)

while a monocular camera is placed behind the vehicle to estimate the pose of the car using the markers (see Fig. 6(a)). Thus, the coordinates of the front checkerboard’s corners are transferred to the rear camera image -using a tri-focal tensor. The ground truth is obtained by removing the car from the field of view of the camera. With such setup we evaluate the accuracy of the entire system; the pose estimation and the image synthesis together. An example of transferred points is available in Fig. 6(b).

A series of 19 sets of images have been acquired with different positions of the car, the computed Euclidean re-projection error is available in Fig. 7. We compared the pose estimation obtained using a checkerboard fixed on the back of the car with our markers’ based approach. From our results, it is clear that the use of the markers leads to a significantly higher accuracy in the pose estimation of the car. Even with the markers the mean error of the transferred points over all the images is 8.2 pixels. Nonetheless, the standard deviation of this error is very small (under 0.5 pixel) which mean that our synthetic image is coherent but misaligned. This transfer error can be attributed to the noise and the numerical errors accumulated along the different steps of our approach (pose estimation, calibration, stereo matching and tri-focal transfer). In practice, such error in the generation of the synthetic image is acceptable. Indeed, considering the size of the original image used for this test (1280×720 pixels), a misalignment of 8

pixels only represents 0.625 % of the image's width. For the desired application it is enough to generate a realistic seamless transparency effect, as it is emphasized in the next section.

5.2 Results in Real Conditions

For these experiments, two cars are utilized, the front car is equipped with the previously described markers in order to estimate the inter-car poses. The cars have been driven within a speed range of 0 to 40 km/h on a campus. Our algorithm is running at 15 frames per second and all the results presented in this section have been computed on-line. Notice that this frame-rate is significantly higher than existing real-time approaches such as [5] which can only process 5 images per seconds without providing a seamless transparency effect (and using a more cumbersome marker setup).

The Fig. 8 shows a representative set of results, in this figure the yellow bounding box is the ROI to be processed while the red dots are the detected corners utilized for the pose estimation. Overall, we can notice that the images are well aligned and allow the driver to have a good overview over the occluded part. Our algorithm works in many challenging situations such as the downhill case (see Fig. 9(e)), while most of the already existing methods [6, 7] cannot deal with such circumstances.

The Fig. 8(f) and (d) show results with far away background and close-by objects. In both cases we are able to obtain an accurate synthetic image of the occluded scene. Other techniques like [6] only work for the ground surface, while [7] relies on a very approximative depth map obtained from a laser range scanner. Therefore they are not able to tackle a large variety of scenarios unlike our method.

During our experiments we also crossed other moving vehicles (see Figs. 8(e) and 9(b)), despite the speed of the cars our algorithm still provides a good image alignment. In the same way, on the Fig. 9(a), our cars are crossing a group of pedestrians.

Even when the vehicles are not aligned, as it is the case in Fig. 9(c) it is still possible to recover a proper image of what is behind the front car. Notice that in some cases the four markers cannot be detected or tracked properly, even in these contexts, the pose estimation remains accurate and efficient. Nonetheless, the proposed approach suffers multiple drawbacks. The main one is probably the limited working distance of the system, indeed, when the cars are spaced by more than 10 m the markers could not be tracked with a sufficient accuracy which leads to a wrong pose estimation. However, this range is usually enough to increase the road safety and to see the dangerous obstacle through the front car. Furthermore, This working distance could be increased with a new design of the markers or by augmenting the resolution or the focal length of the rear car camera.

Note that, the transparency effect cannot be applied on the lower part of the front car since this part is not visible by the stereo-vision system (this is



Fig. 8. Real time results in real condition, (first row) markers detection and pose estimation, (second row) Seeing-through results

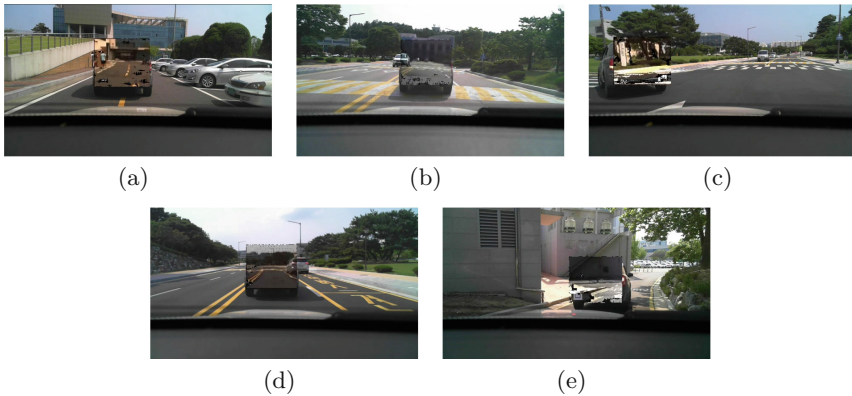


Fig. 9. Sample of results obtained with our approach

actually the part located under the car). This is the reason why this area is not fully recovered in our experiments.

6 Conclusion

In this paper we presented a novel approach to see through the front vehicle in real time using a very low cost setup consisting of only three cameras and a standard Wi-Fi connection. Unlike the already existing methods, our approach provides a seamless transparency effect which respects the geometry of the scene. Furthermore, our algorithm has been designed to reduce the quantity of data to be transferred between the vehicles which allows to process up to 15 frames per second. Moreover, our marker-based approach is very efficient within a range of 1 to 10m and even when the vehicles are static.

We performed multiple tests with real data which underlines the efficiency of our approach in term of accuracy. Furthermore, a large set of real experiments have been done. In these experiments it is clear that our system is working in practical situations. However, our method can be improved in multiple ways. For instance, the integration of an appropriated display such as a HUD or Google glasses is an interesting direction. For a realistic deployment of this seeing through system, a marker-less approach has to be considered. Our current researches focus on marker-less pose estimation and SLAM-based approach to avoid the use of markers. Finally, our method can easily be declined for night vision by replacing the markers by IR LEDs.

Acknowledgements. This research was supported by the Shared Sensing for Cooperative Cars Project funded by Bosch (China) Investment Ltd.

The first author was supported by Korea Research Fellowship Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning (2015H1D3A1066564).

References

1. Naja, R. (ed.): *Wireless Vehicular Networks for Car Collision Avoidance*. Springer, New York (2013)
2. Clarke, D., Ward, P., Jones, J.: Overtaking road-accidents: differences in manoeuvre as a function of driver age. *Accid. Anal. Prev.* **30**(4), 455–467 (1998)
3. Olaverri-Monreal, C., Gomes, P., Fernandes, R., Vieira, F., Ferreira, M.: The see-through system: a VANET-enabled assistant for overtaking maneuvers. In: *IV* (2010)
4. Gomes, P., Vieira, F., Ferreira, M.: The see-through system: from implementation to test-drive. In: *VNC* (2012)
5. Gomes, P., Olaverri-Monreal, C., Ferreira, M.: Making vehicles transparent through V2V video streaming. *IEEE Trans. Intell. Transp. Syst.* **13**(2), 930–938 (2012)
6. Kim, S., Qin, B., Chong, Z., Shen, X., Liu, W., Ang, M.H., Frazzoli, E., Rus, D.: Multivehicle cooperative driving using cooperative perception: design and experimental validation. *IEEE Trans. Intell. Transp. Syst.* **16**(2), 663–680 (2015)
7. Li, H., Nashashibi, F.: Multi-vehicle cooperative perception and augmented reality for driver assistance: a possibility to see through front vehicle. In: *ITSC* (2011)
8. Horaud, R., Dornaika, F.: Hand-eye calibration. *Int. J. Robot. Res.* **14**(3), 195–210 (1995)
9. Lébraly, P., Ait-Aider, O., Royer, E., Dhome, M.: Calibration of non-overlapping cameras-application to vision-based robotics. In: *BMVC* (2010)
10. Bleser, G., Pastarmov, Y., Stricker, D.: Real-time 3D camera tracking for industrial augmented reality applications (2005)
11. Caron, G., Marchand, E., Mouaddib, E.M.: 3D model based pose estimation for omnidirectional stereovision. In: *IROS* (2009)
12. Petit, A., Marchand, E., Kanani, K.: Augmenting markerless complex 3D objects by combining geometrical and color edge information. In: *ISMAR* (2013)
13. Kato, H., Billinghurst, M.: Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In: *IWAR* (1999)
14. Shi, J., Tomasi, C.: Good features to track. In: *CVPR* (1994)

15. Lepetit, V., Moreno-Noguer, F., Fua, P.: EPnP: an accurate $o(n)$ solution to the PnP problem. *Int. J. Comput. Vision* **81**(2), 155–166 (2009)
16. Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge Univ Press, New York (2003)
17. Hirschmuller, H.: Stereo processing by semiglobal matching and mutual information. *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(2), 328–341 (2008)