Front Vehicle Blind Spot Translucentization Based on Augmented Reality

Che-Tsung Lin, Yu-Chen Lin, Long-Tai Chen
Mechanical and Systems Research Laboratories
Industrial Technology Research institute
Chutung, Hsinchu, Taiwan 31040,R.O.C
{alexlin,davidlin,ltchen}@itri.org.tw

Yuan-Fang Wang
Department of Computer Science
University of California
Santa Barbara, California 93106
yfwang@cs.ucsb.edu

***Abstract*—Recently, WAVE/DSRC has become an attractive technology for vehicular safety applications. Vehicles with WAVE/DSRC devices can communicate with their neighboring vehicles to exchange information to achieve collaborative safety. This paper proposes a new vehicle blind spot elimination system which utilizes the on-board videos captured from other vehicles and the host vehicle. The preceding vehicle which fully or partially blocks the field of view of the host vehicle could be translucentized in the video captured by the host vehicle and the driving environment of the front vehicle could be then visually checked by the host driver.**

I. Introduction

The statistics from NHTSA (National Highway Traffic Safety Administration) show that, in the U.S, 31% of traffic accidents are due to rear-end collisions. Often times, a rear-end collision happens when a front vehicle suddenly slows down or stops, giving the following vehicles no warning and little time to react. Traditional passive safety systems could only reduce injuries or casualties. On the other hand, active vehicle safety systems could prevent the potential hazard from happening.

Active vehicle safety systems can roughly be classified as autonomous and collaborative schemes. In autonomous solutions, many collision warning systems [1], [2], [3], [11] have been proposed. For instance, a vehicle can detect its distance to another vehicle via sensors, such as laser, radar, and camera. Then, emergency events could be prevented using the perceived information. However, even though a vehicle is equipped with such a system, the driver still could not respond quickly if a preceding vehicle fully or partially blocks the field of view of the host vehicle and suddenly slows down or stops. As to the collaborative solutions, location information of vehicles is periodically exchanged to prevent potential danger in advance [9], [10]. It is difficult for the drivers to sense the immediate danger electronically because human beings tend to believe in what they see with their own eyes. Hence, we believe that enhancing the visual feedback to the driver is the key to improved safety.

In this paper, we propose a new vehicle blind spot elimination system which utilizes the on-board videos captured from multiple vehicles to make vehicles on the road translucent. The particular scenario involves two vehicles, one following the other. As a result, the view of the back vehicle is often partially blocked by the front vehicle. If both vehicles are equipped with a front-looking camera and a V2V communication device which allows the exchange of the video streams, it is then possible to replace the image of the front vehicle in the video stream of the back vehicle by what the front vehicle sees, and thus eliminate the blind spot created by the front vehicle in the back cameras video stream.

This data analysis and fusion paradigm is best understood by examples, and four are shown in Fig. 1 (one per row). The left column of Fig. 1 shows what the front vehicle sees, and the middle column shows what the back vehicle sees. Depending on the separation between the front and back vehicles, the view of the back vehicle is partially blocked by the front vehicle. In some cases, the blockage can severely limit the ability of the driver of the back vehicle to interpret the road condition ahead. On the right column, we show that by using our sensor data fusion algorithm, it is possible to make the front vehicle "translucent" in the back video frame, and hence, provide a much better visual feedback to the driver of the back vehicle of the road ahead.

The proposed algorithm uses image analysis to achieve sensor data registration and fusion, and *does not rely on any other external sensors such as GPS and gyroscope*. Furthermore, the visualization shown in the right column of Fig. 1 is from the point of view of the host driver, not simply displaying the videos of the front, blocking vehicle to the host driver. Below, we will describe our algorithm in detail.

II. Technical Rationale

The gist of the algorithm is to use image analysis methods to infer the relative pose between the two cameras, identify the image location of the front vehicle in the back image frame, and blend the front image content into the back image around the front vehicle location. The process makes use of feature correspondences extracted and identified in these images—that is, objects seen by both cameras. Such feature correspondences enable the following computations: (1) sensor registration: ascertain the relative pose (rotation and translation) of the two cameras, (2) vehicle localization: determine the locations of the epipoles, or the projected location of the (front, back) camera's optical center in the image plane of the (back, front) camera, and (3) data fusion: compute both a mapping equation and the size of the mapping region where pixel values of the front image are blended into the corresponding pixels in the back image. These steps are discussed in more detail below.

a) Feature detection: For an object in an image, representative points on the object can be extracted to provide a characteristic description of the object. This description, extracted from an image, can then be used to locate the object in another image. To perform reliable object matching, it is important that the features extracted from an image be

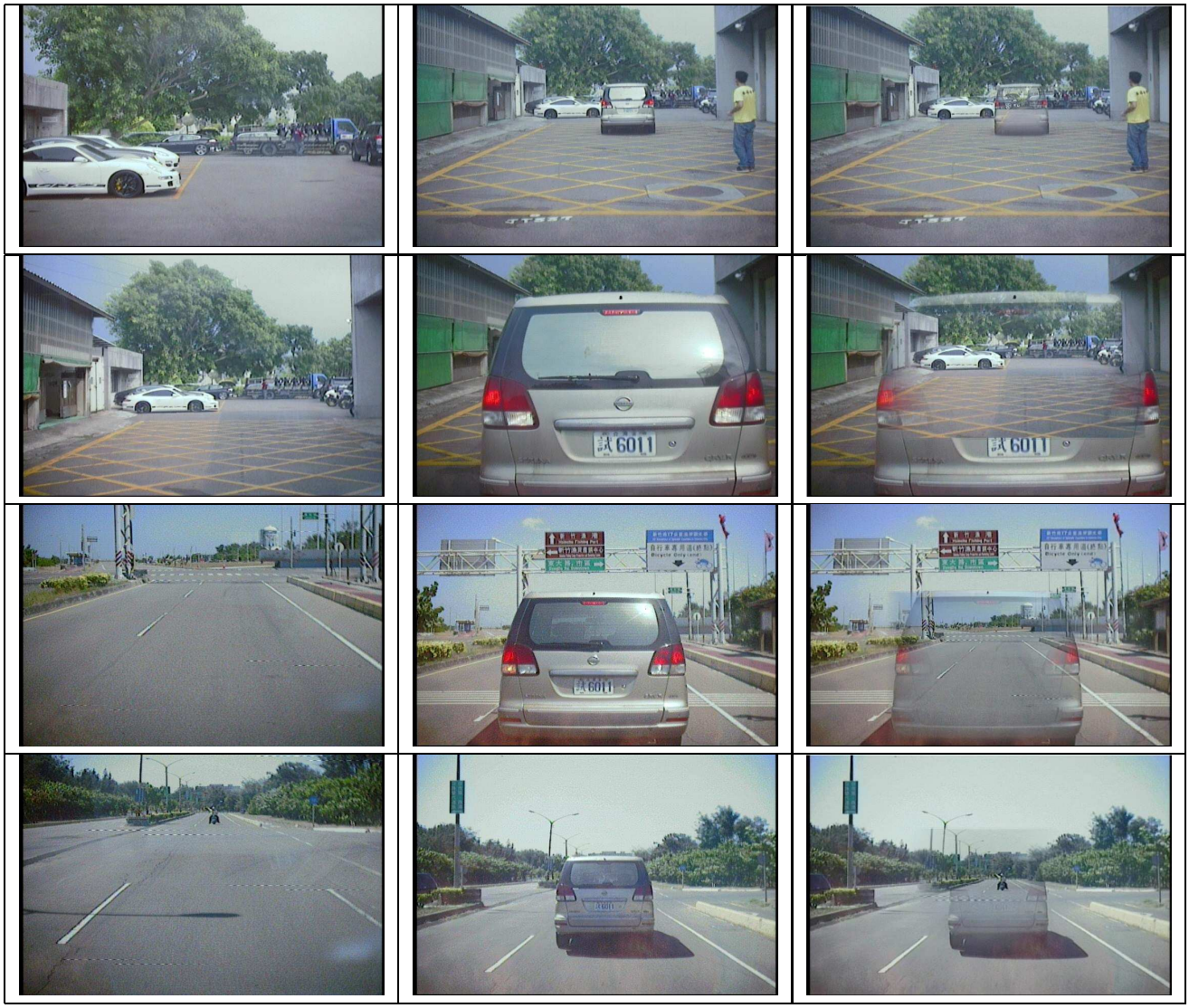

Fig. 1. Left: image seen by the front camera, center: image seen by the back camera, and right: the back image with blended-in content from the front image.

detectable in another image even with changes in image scale, noise and illumination.

One popular feature detection and description method is due to David Lowe. Lowe's SIFT method [7] transforms an image into a large collection of feature vectors, each of which is invariant to image translation, scaling, and rotation, partially invariant to illumination changes, and robust to local geometric distortion. Key locations are defined as maxima and minima of a difference of Gaussian (DoG) function applied in a scale-space to a series of smoothed and re-sampled images. Low contrast candidate points and edge response points along an edge are discarded. Dominant orientations are assigned to localized keypoints. These steps ensure that the keypoints are more stable for matching and recognition. SIFT descriptors robust to local affine distortion are then obtained by considering pixels around a radius of the key location, blurring and resampling of local image orientation planes. We have used a public-domain SIFT implementation (www.vlfeat.org) for this purpose.

b) Feature matching: SIFT scheme [7] uses a 128-element-long feature descriptor that characterizes the gradient pattern in a properly oriented neighborhood surrounding a SIFT feature location in a way that is (semi-)invariant to incidental environmental changes in lighting, viewpoint, and scale. Note that we do not know the relative pose of the front and back cameras as no external sensor is used. Hence, matching SIFT feature descriptors in two images is a 2D search based on the similarity of the descriptors only. Algorithmically, we use a brute-force $O(n^2)$ scheme to match features detected in the front and back images. A potential match must satisfy two criteria: the pairing itself must be of a high quality and it must be much better than all other possible matings—meaning that the match should not be ambiguous. We compute the Euclidean distance between two SIFT feature vectors as the matching score and require that the ratio of the matching scores of the best pairing and the second best to be less than 0.8, a constant suggested in [7]. The left column of Fig. 2 shows the resulting matches for the four examples shown in Fig. 1. Note that erroneous pairings do exist based only on feature

similarity. We will attempt to filter out these incorrect matches using geometrical constraints discussed below.

c) RT computation: Based on the feature correspondences identified in the previous step, we compute the relative pose (R: rotation and T: direction of translation) between the two cameras. The inference process imposes the epipolar constraint in terms of a Fundamental Matrix relation: $p'Fp = 0$, where F is the Fundamental Matrix [5]. As shown in Fig. 3, P represents the 3D feature location while p and p' the projected 2D feature locations in two images (or p and p' form a corresponding pair). O and O' denote the optical centers of the two cameras, and Π and Π' are the image planes.

Geometrically, it is easy to see that points O, O', p, p' and P all lie on the same plane (the epipolar plane). Hence,

$$O'p' \cdot (O'O \times Op) = 0. \tag{1}$$

Denote the camera's intrinsic matrix [5] as K and K' for the two cameras; we can convert p and p' from pixel coordinates to real-world coordinates as $K^{-1}p$ and $K'^{-1}p'$. If we denote the movement of the camera between the two frames as R and T, then quite obviously $O'O = T$ and Op expressed in the primed frame is then

$$Op = RK^{-1}p + T. \tag{2}$$

$$\begin{array}{rcl} (K'^{-1}p')^T(T \times (RK^{-1}p + T)) & = & 0 \\ p'^T K'^{-T}(T \times RK^{-1}p) & = & 0 \\ p'^T(K'^{-T}T \times RK^{-1})p & = & 0 \\ p'^T Fp & = & 0 \end{array} \tag{3}$$

Each corresponding feature pair identified in step 2 above then provide one constraint on F. With enough such feature correspondences, we can solve for F, and hence T and R. The process can be based on either the Calibrated Five Point Algorithm of Nister [8] or the Normalized Eight Point Algorithm of Hartley [6].

While the names of the inference algorithms refer to the minimum numbers of pairs of matched image features in two views that are needed for deducing the camera's motion parameters, in reality, we can often match significantly more features than just five or eight. Furthermore, matching results are necessarily imprecise due to noise and image quantization, and catastrophic failure in erroneous pairing assignments does happen occasionally as shown in Fig. 2. To improve the robustness in camera motion inference, we use a nonlinear selection and filtering strategy called RANSAC [4] to better condition the feature matching process by imposing the epipolar constraints.

The essence of the RANSAC selection process is to repeatedly apply 5-point or 8-point computation to a small subsets of randomly selected "seed" correspondences, in hope that at least one set of seed correspondences were not corrupted by bad, outlier correspondences. Such outlier-free seeds will lead to an F that produces small residual errors in $|p'Fp|$ for most inlier correspondences. Hence, outlier correspondences are identified and filtered out as those with unreasonably large residual errors. The best seed selection corresponds to the one that produces the minimum median $|p'Fp|$ residual error. One final run of the 5-point/8-point algorithm is then performed using all inlier correspondences in a least-squared sense.

d) Estimation of the positions of the epipoles: Geometrically speaking, Fp represents the epipolar line l' in the primed frame [5] or $l' = Fp$ because the linear epipolar relation $p'Fp = 0$ implies that p' lies on Fp. As shown in Fig. 3, all such epipolar lines pass through the epipole e' in the primed frame, and hence, $e'^T Fp = (e'^T F)p = 0$ for all p. Hence, e' must be the left null vector of the Fundamental matrix F. Similarly, one can show that $l = F^T p'$ or $e^T F^T p' = p'^T Fe = p'^T(Fe) = 0$ for all p'. e is therefore the right null vector of the Fundamental Matrix. We can solve for e and e' using the standard Singular Value Decomposition (SVD) of F [5].

The results of steps c and d above are shown in the right columns of Fig. 2 with the estimated epipole positions marked in red. As can be seen that the epipole positions look reasonable (inside the image of the front vehicle) and many erroneous feature pairings from step b were filtered out.

e) Estimation of the sensor data fusion parameters: This step estimates both the geometric configuration and functional form necessary for achieving sensor data fusion. Firstly, we need to know three pieces of information related to the geometric configuration, namely, the location, size, and shape of the fusion region. There are at least two distinct mechanisms to determine this region in the back camera's frame. One possibility is that some vehicle detection algorithm is used to locate the front vehicle in the back image. The fusion region can then be the region identified by the vehicle localization algorithm, with the goal of replacing pixels inside the vehicular region by the corresponding pixels seen in the front image.

In the absence of such a solution, some educated guess of the fusion region must be made. Intuitively speaking, the epipole in the back image gives the position of the front camera's optical center. This location, if inferred correctly, must be inside the front vehicle. Hence, barring evidence saying otherwise, a reasonable choice is to center the fusion region around the epipole. A circular shape is often assumed to avoid introducing color blending artifacts. The size (or radius) of the blending region is chosen based on the size of the regions surrounding the epipoles that are devoid of matched features (often implying that the two cameras are seeing different objects, and hence, no matched feature pairs can be identified).

Secondly, once the fusion region in the back frame is identified, we need to compute a functional form, $p' = f(p)$, to map pixels from the front image to the back one. While such a relation in general can be highly nonlinear and be different for different image neighborhoods, we feel that it is not technically feasible to derive an accurate relation given the stringent time constraint (be able to present the fused images in real time to the driver) and a limited amount of data. Nor is such an accurate mapping relation necessary, as our experiments showed that we can roughly approximate the mapping relation in a global, linear form in polar coordinates to achieve visually appealing color fusion results (Fig. 1).

Refer to Fig 4(a), the front-and-back epipolar geometry dictates that the epipoles form a matching pair and serve naturally as the blending centers. This is because the optical axes of the two cameras are perfectly aligned regardless of the depth of the 3D object viewed at the epipole location. Furthermore, the epipolar geometry is most conveniently described in a polar system centered around the epipoles, as shown in Fig. 4(a).

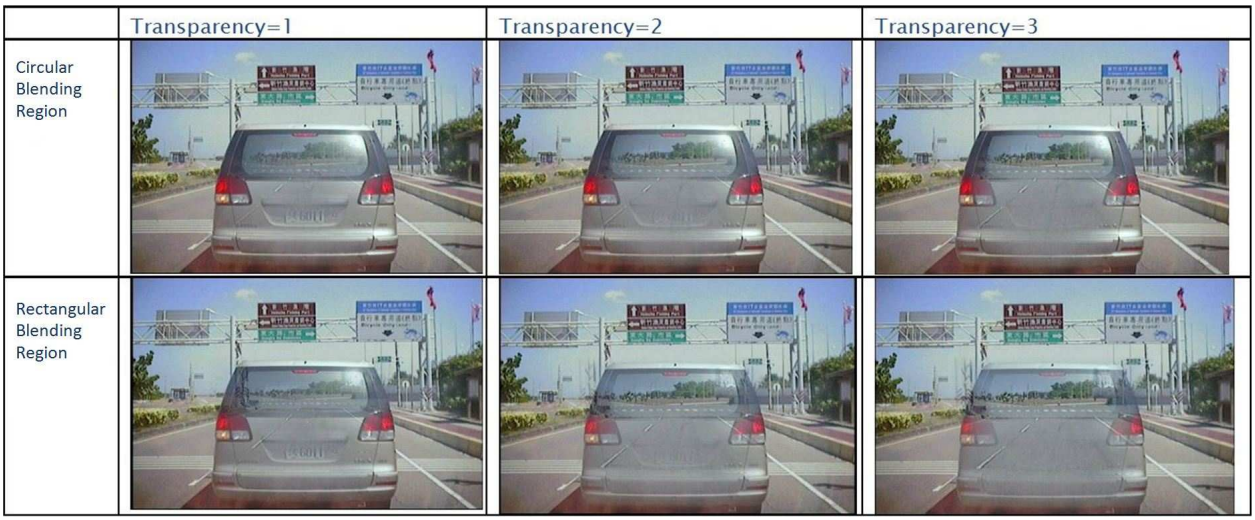

Fig. 5. Different blending effects by varying blending transparency and shape parameters.

Therefore, we model the pixel mapping relation as a global, linear relation in polar coordinates centered at the epipoles, or we try to infer two linear equations, $(p' - e') = \mathcal{R}(p - e)$ and $(p' - e') = \mathcal{A}(p - e)$—one for polar radial $\mathcal{R}$ and the other for polar angle $\mathcal{A}$, that map a point p in the front image to a point p' in the back image. That is, we translate the origin of each image plane into its respective epipole and then convert the Cartesian coordinates $(p - e)$ into the polar coordinate (ρ, θ). Using the corresponding features (obtained from steps a to d) in the front and back images, we perform a best linear fit to derive $\mathcal{R}$ and $\mathcal{A}$. Again, while such a global, linear fit cannot account for local depth variation, it works well for performing global color blending.

f) Color blending: The final step is to modify the colors of the pixels in the back image which depict the front vehicle with the corresponding pixels seen by the front camera.

The blending process is applied to a circle (or a box) of confusion centered at the epipole in the back image. The radius of the circle or the size of the box is estimated as a percentage of the region around the epipole in the back image that is devoid of feature correspondences or is the size of the vehicular region reported by a vehicle localization algorithm. The blending mixture is controlled by a transparency parameter that makes the front vehicle more or less apparent inside the region of confusion. The effect of these two parameters (shape and transparency) combined is best illustrated in Fig. 5. As can be seen that with increased transparency (from left to right), the vehicle becomes less visible. Comparing the results on the same column (with the same transparency parameter), the rectangular box hides the vehicle better along the vehicle's boundary.

III. Experimental Results

We have equipped two minvans with forward-looking cameras. A video capturing system was also installed that allowed the on-board video to be captured and stored for later analysis. The two vehicles were driven in a front-and-back following configuration on many different roads (inside the ITRI campus and on major highways in Taiwan). We then applied the proposed algorithm for feature analysis, camera pose calibration, and color blending. Sample results are shown in Figs. 1, 2, and 6.

IV. Concluding Remarks

In this paper, we introduce a computer-vision and augmented-reality based algorithm to eliminate blind spots in on-board videos.

References

[1] A. Bensrhair and A. Bertozzi and A. Broggi and A. Fascioli and S. Mousset and G. Toulminet. Stereo vision-based Feature Extraction for Vehicle Detection. In *Proceeding of the IEEE Intelligent Vehicles Symposium*, pages 465–470, Jun 2002.

[2] A. P. Wang and J. C. Chen and P. L. Hsu. Intelligent CAN-based Automotive Collision Avoidance Warning System. In *Proceeding of the IEEE International Conference on Networking, Sensing and Control*, pages 146–151, Mar. 2004.

[3] C. C. Lin and C. W. Lin and D. C. Huang and Y. H. Chen. Design a Support Vector Machine-based Intelligent System for Vehicle Driving Safety Warning. In *Proceeding of the IEEE Conference on Intelligent Transportation System*, pages 938–943, Oct. 2008.

[4] M. Fischler and R. Bolles. RANdom Sample Consensus: A Paradigm for Modeling Fitting with Application to Image Analysis and Autoamted Cartography. *Communications of ACM*, 24:381–395, 1981.

[5] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Camridge, MA, 2003.

[6] R. I. Hartley. In Defense of the Eight-Point Algorithm. *IEEE Trans. Pattern Analy. Machine Intell.*, 19, 1997.

[7] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision*, 60:91–100, 2004.

[8] D. Nister. An Efficient Solution to the Five-Point Relative Pose Problem. *IEEE Trans. Pattern Analy. Machine Intell.*, pages 756–770, 2004.

[9] Q. Xu and T. Mak and J. Ko and R. Sengupta. Vehicle-to-Vehicle Safety Messaging in DSRC. In *Proceeding of the 1st ACM International Workshop on Vehicular Ad Hoc Networks*, Oct 2004.

[10] T. E1Batt and S. K. Goel and G. Holland and H. Krishnan and J. Parikh. Cooperative Collision Warning Using Dedicated Short Range Wireless Communications. In *Proceeding of the 3rd ACM International Workshop on Vehicular Ad Hoc Networks*, pages 1–9, Sep 2006.

[11] Y. Ying and W. Ximing. The Research of Collision Avoiding System Based on Millimeter Wave and Image Processing Technique. In *Proceeding of the International Conference on Computational Intelligence and Security*, pages 1789–1792, Nov 2006.

Fig. 2. Left: feature correspondences for the four examples in Fig. 1 from similarity matching. Right: feature correspondences for the four examples from similarity matching and epipolar pruning.

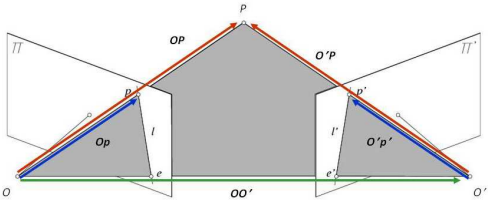

Fig. 3. A typical stereo configuration.

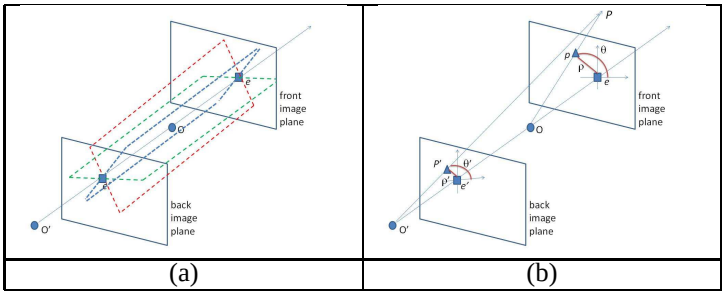

Fig. 4. A front-and-back stereo configuration.

Fig. 6. Left: image seen by the front camera, center: image seen by the back camera, and right: the back image with blended-in content from the front image.