Enhancing Vehicular Safety in Adverse Weather using Computer Vision Analysis

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, CA 93106
yfwang@cs.ucsb.edu

Abstract— the goal of the project is to design intelligent and robust image-processing and augmented-reality algorithms for driver assistance and enhanced vehicular safety. In particular, the focuses were two-fold: (1) realizing the abilities to identify and localize in a vehicle's on-board video the sweeping windshield wipers during raining days and (2) designing and implementing an in-painting technique to remove the image of the windshield wipers and replace it with the corresponding pixels (not blocked by the wipers) from an adjacent frame.

I. I. INTRODUCTION

Advances in video technology have enabled its wide adoption by the auto industry. For examples, many vehicles today are equipped with backup and side-looking cameras that allow the driver to easily monitor the traffic around the vehicle for enhanced safety. V-to-V communication (either directly or through relay stations strategically positioned along highways and road junctions) facilitates the exchange and sharing of video information among multiple vehicles for better cognizance of the surrounding road conditions, hazardous incidents, and traffic accidents.

In such V-to-V communication, a vehicle's onboard video can be processed automatically, say, to remove undesirable motion blur and pixel blockage before retransmission to others for heightened situation awareness and driver assistance. This research is centered on such a video refinement scheme, namely in detecting and localizing wiper pixels and replacing them with the corresponding unblocked pixels from an adjacent non-wiper frame in the onboard video to improve the visual feedback to the drivers. We address two problems here: that of wiper detection and localization and, once the presence of a wiper is identified, to mask off the wiper pixels with suitable unblocked, non-wiper pixels from an adjacent video frame for enhanced viewing.

This process is best illustrated by an example. As shown in Figure 1, during the raining days, many video frames may have the wiper presence that blocks out the road and vehicles on the road (Figure 1 left column). If we can detect and localize the wiper pixels in such a wiper frame (Figure 1 middle column) and find a suitable adjacent video frame without the wiper presence, we can then replace those wiper pixels in the wiper frame with the corresponding pixels in the adjacent non-wiper frame (Figure 1 right column) to enhance the viewing feedback to other drivers through V-to-V communication.

Figure 1(a) a video frame with the wiper presence, (b) the wiper frame with the wiper detected and localized automatically by our computer program (wiper pixels are marked in red), and (c) wiper pixels in the wiper frame are replaced by the corresponding pixels in the adjacent non-wiper frame to provide better viewing of the surrounding.

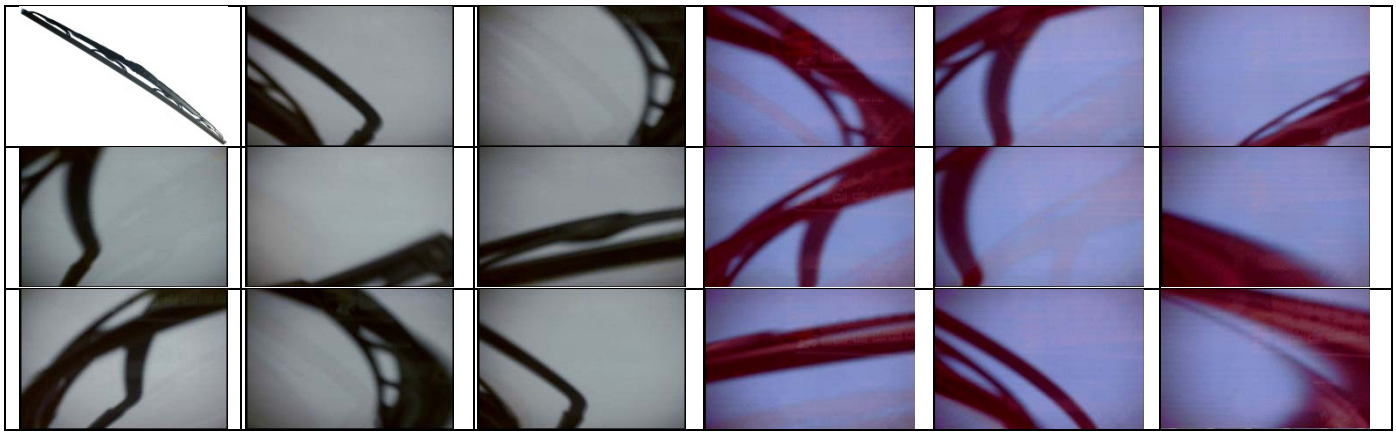


Figure 2 Upper-left corner: a standard "coat-hanger" wiper, and the rest: different ways a standard wiper can appear in an onboard video (the pictures were taken against a uniform background for clarity).

Note that while in-painted regions in a wiper frame may not faithfully depict objects on the road at that particular time instance - as the in-painted pixels are extrapolated from an adjacent, non-wiper frame - the improved visual feedback to the driver is useful as wiper blockage can be significant as shown in Figure 1. Furthermore, as will be shown later, a wiper frame and the adjacent non-wiper frame used for masking the wiper pixels are separated by tens of milliseconds at most. Hence, the changes of road conditions are often negligible in such cases. We believe that such a wiper-masking system can be of great assistance to the driver in improving visualization and situation awareness.

In this paper, we describe our technical formulation and present some preliminary experimental results to validate the approach. We are currently implementing the algorithm on an on-board DSP, and the timing and performance results will be described in an ensuing paper.

II. TECHNICAL RATIONALE

As mentioned before, the wiper elimination algorithm comprises two steps: (1) wiper classification and localization, and (2) wiper pixel masking and in-painting. We describe these two steps below:

A. *Wiper Classification and Localization*

We surmise that there are at least two possible approaches to the wiper localization and tracking problem: top-down, model-guided and bottom-up, image-directed. A top-down approach creates a graphic model of the wiper, and the model is then animated to match the sweeping motion of the real wiper. One advantage of a model-guided formulation is that well-established tracking frameworks, such as the Kalman Filter and the Particle Filter [1,2], are readily applicable. That is, essential model parameters (shape, pose, etc.) can be extracted from the graphic wiper model to form a state-space representation. The said representation can then be evolved over time using the standard "initialization-prediction-observation-correction" processing cycle of a Kalman or Particle Filter to track the movement of the wiper in a vehicle's onboard video.

While such a top-down, model-guided approach appears conceptually sound, the execution of such a scheme in real videos presents many challenges. To name a few:

(1) While a simple abstraction of a wiper is a narrow, highly elongated, and maybe slightly curved rectangle, its shape is actually much more complicated. The standard wiper blades have a "coat-hanger" shape at rest, but the shape can deform drastically at various stages of the sweeping actions as illustrated in Figure 2. To accurately model such a large degree of deformation, sophisticated and elaborate graphic models are needed to capture the dynamic shape change. As more variances of the conventional wiper design are introduced, new graphic models are needed to describe them. Hence, an anatomical model may be particular to the wipers of a specific make and model, making model generalization an expensive and manual process.

(2) Conventional wipers have a small, fixed number of sweeping speeds (usually low, median and high). With the newer intermittent and the rain-sensing wipers, the sweeping speed can assume a wide, continuous range to complicate the computer analysis. Furthermore, a wiper's sweeping action and the camera's shutter are *not* synchronized in any way. Our observation is that even with the wiper in a low-speed setting; its movement across the windshield is too fast to easily defeat automated tracking. In Figure 3, we show some typical sequences of an activated wiper with the speed at a low setting**.** The sampling rate is 30 frames per second. What we observe is that (a) a wiper does not stay visible for more than three or four consecutive frames, (b) the wiper can often travel over half an image length or more and (c) the shape of the wiper can change drastically from one frame to the next (over 1/30th second). In fact, we know of no tracking algorithm that can reliably track movements over such a large distance and exhibiting such a large deformation, without the guidance of a highly accurate domain model.

Therefore, we have decided to abandon the top-down, model-based formulation in favor of a bottom-up, image-directed one. We believe that such an image-based approach enjoys a number of advantages over a model-based one in that

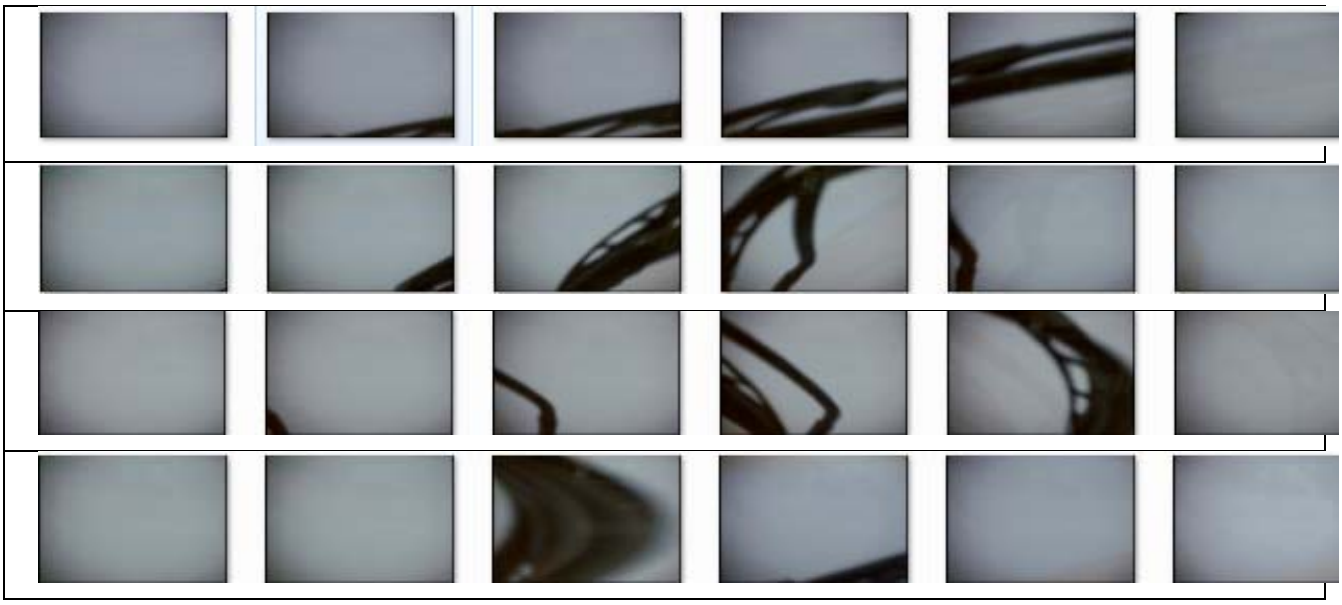


Figure 3 Each row represents one continuous, yet ephemeral, presence of the wiper in the onboard video.

it is much simpler to formulate, readily generalized to wipers of all makes and models, and with a potential for fully automated deployment.

The main idea is to exploit the principal component analysis [3]. The approach was motivated by the various "eigen"-representations made popular in computer vision (e.g., eigen-faces [4]). As the camera is rigidly affixed to the rear-view mirror and the relative position of the camera and the windshield does not change, ideally a wiper should have a consistent appearance when imaged at the same location on the windshield. If we were to observe the wiper at many possible locations on the windshield (using training images or training videos), it is then possible to detect the presence of and localize the wiper in other videos by a similarity search.

To translate the above observation into an efficient and robust classification scheme, we need to pay attention to the following implementation issues:

***Efficiency*:** A brute-force method may compare a test image with each and every wiper image in the training set. This will then limit the number of training images that can be used. Wiper comparison can be done much more efficiently by organizing the training wiper images effectively to eliminate redundancy. Hence, we construct an "eigen"-wiper space to represent all wipers in the training images. The representation is advantageous in that (1) we do not use images at the full native resolution. For the relatively uncomplicated geometrical shape that a wiper assumes, down-sampling reduces the pixel count at no material loss on the recognition accuracy, and (2) the training wiper images are vectorized and collectively represented by a set of orthogonal basis vectors (or "eigen" wipers). While the full dimensionality of a vectorized wiper image is the number of pixels - a very large number even with down-sampling, together all these vectorized wipers occupy only a small subspace in this large vector space. Hence, instead of comparing pixel values, we compare the wipers' projection coefficients in the vector subspace using a greatly reduced effort.

***Robustness*:** As the camera shutter and the movement of a wiper are *not* synchronized, the wiper does not appear in just a small number of "canonical" positions on the windshield. To observe all possible wiper appearance will require a very large number of training images. We institute two mechanisms to extrapolate from the observed wiper locations to likely wiper locations not observed in the training images: (1) a short-range extrapolation: this is accomplished by smoothing the training wiper images and thus enlarging the "footprint" of a wiper to better predict the wiper presence in close-by locales, and (2) a long-range prediction: using a rigid-body transformation, a wiper image is aligned with its most similar counterpart in the training set. The alignment facilitates the prediction to the likeliness of two wipers brought together artificially.

In more detail, the wiper classification algorithm comprises three phases: training, validation, and deployment. The training phase is for constructing the eigen wiper space using a set of training images containing a wiper. The validation phase uses two types of labeled images: positive ones with wiper presence and negative ones without. The goal is to learn the best system dichotomy parameters for classifying unseen images into the wiper and non-wiper classes. The deployment phase uses such a classification system in real-world scenarios.

Due to the page limit, we will not describe our wiper segmentation algorithm in detail. Suffice it to say that a wiper has a distinct color signature (much darker than the surroundings). We use a K-mean clustering algorithm to separate images into a small number of clusters and identify a cluster with a sufficiently large footprint and a dark signature as a potential wiper region (sample segmentation results are shown in the left columns of Figure 4*).* Note that we do not use the shape characteristics in this initial segmentation stage. This is because, as shown in Figure 2, the wiper shape can change

drastically and we use our eigen-wiper scheme to capture such shape variation.

Training: The training phase comprises the following major processing steps:

(1) A wiper localization program (based on K-mean clustering) is first applied for localizing the wiper regions in an image.

(2) The resulting image is then converted into a binary mask with a pixel value of 1 representing wiper pixels and 0 representing non-wiper pixels. This binary conversion is essential as wiper pixels can assume varying color and intensity values in different images. Furthermore, the non-wiper, or background, pixels can depict all kinds of scene objects. For the eigen-representation to be representative and repeatable, such random variation need be eliminated.

(3) Wiper images are then convolved with a Gaussian kernel to smooth out and enlarge the footprint of the wiper region. This step is used to resolve small misalignment of wipers and is particularly useful when the footprint of the wiper is small, and hence, even a slight misalignment can be problematic.

(4) The resulting images were then down-sampled to reduce the pixel count (from 720 by 480 to 90 by 60 in our setup).

(5) The down-sampled images are vectorized by performing a linear, sequential scan of all pixels (a vector of 5,400 long).

(6) The mean wiper vector is computed as the average of all vectorized training wipers and the mean is subtracted from all wiper vectors.

(7) All such vectorized training images are then collected, column-wise, into an observation matrix **O**. An optional size parameter can be used in this step to eliminate training images where the size of the wiper region is too small.

(8) The singular value decomposition of the observation matrix is performed, or $\mathbf{O} = \mathbf{U}\mathbf{D}\mathbf{V}^T$. The columns of **U** represent eigen wipers, the diagonal matrix **D** records the significance (or importance) of each eigen wiper in a non-increasing order, and the matrix **V** records the projection coefficients (coordinates) of the training images in the eigen wiper space.

(9) An optional accuracy parameter can be used to eliminate eigen wipers whose singular values in the **D** matrix is smaller than a certain threshold.

Validation: The validation phase comprises the following major processing steps using labeled wiper images, both positive and negative ones. For each test images, two matching scores are computed; the first matching score is used to select the closest neighbor in the set of training images and is based on the separation of a test image and training images in the eigen wiper space. After the closest neighbor, or the best match, of a test image is found, we attempt to align the test image and its best match before we compute a second matching score, based on the amount of pixel overlap in the image plane. The second matching scores of all labeled positive and negative images are then used to determine the optimal dichotomy threshold. The major validation steps are summarized below:

(1) Steps 1-6 above are applied to each test image.

(2) The vectorized test image (**I**) is projected onto the eigen wiper space with the projection coefficient as $\mathbf{I}^T\mathbf{U}$.

(3) The likeliness or similarity of **I** to each training image can be computed two ways: either as the inner product (similarity in angle alignment) or as the Euclidean distance (similarity in position alignment). More specifically, if n vectorized training images are used and each one is represented as a vector of dimension p, then **U** is a matrix of dimension p by n, **D** is n by n, **V** is n by n, and $\mathbf{I}^T\mathbf{U}$ 1 by n. $\mathbf{D}\mathbf{V}^T$, an n by n matrix, records the coordinates of the training images in the eigen space in the columns.

Angle alignment is computed as $\{\mathbf{I}^T\mathbf{U}\}\{\mathbf{D}\mathbf{V}^T\}$. The quantity $\{\mathbf{I}^T\mathbf{U}\}\{\mathbf{D}\mathbf{V}^T\}$ is a 1 by n vector and records the inner product of **I** with all **n** training images. We use the notation {**.**} to denote the "normalized" vector or matrix quantities. A normalized vector is a vector of norm 1, and a normalized matrix is one where each column is of norm 1. Position alignment is computed this way: we create an n by n matrix **R** by duplicating the n by 1 vector $\{\mathbf{I}^T\mathbf{U}\}^T$ n times column-wise. As $\{\mathbf{I}^T\mathbf{U}\}^T$ and $\mathbf{D}\mathbf{V}^T$ record the projection coordinates of the test and training images, respectively, in the eigen space, the Euclidean distances between **I** and all training images are then the diagonal of the following matrix $((\mathbf{R}-\mathbf{D}\mathbf{V}^T)^T(\mathbf{R}-\mathbf{D}\mathbf{V}^T))$.

(4) The training image assumes the best matching score (either the largest inner product or the smallest Euclidean distance) with a test image is chosen as the most probable wiper placement in the test image.

(5) An alignment operation is then performed to align the wiper region in the test image with that in the best-match training image. A rigid-body transformation is applied to translate the wiper region so that the centroids of the two wiper regions in the testing image and the best-match training image coincide. The wiper region in the test image is further rotated so that the principal axes line up with those of the wiper region in the best-match training image.

(6) The second matching score is computed as the ratio of the pixel counts of the overlapped region between the aligned test wiper and its best-match counterpart over the average pixel counts of the two wipers.

(7) Two histograms of the matching scores, one for the positive samples and the other for the negative samples, are studied to determine a threshold for wiper/non-wiper classification. This step selects the valley between the two histograms as the threshold by minimizing the misclassification error.

Deployment: The deployment phase comprises the following major processing steps:

(1) Steps 1-6 in the validation phase are applied.

(2) An image is classified as either a wiper image or a non-wiper image based on the threshold determined in the validation step.

B. *Wiper Pixel Masking and In-painting*

Once the presence of a wiper is identified in an image, we would like to replace (or mask off) those wiper pixels with the corresponding non-wiper pixels from an adjacent non-wiper frame. This is a process called in-painting.

To replace blocked, wiper pixels in a wiper image with some unblocked, non-wiper pixels in an adjacent non-wiper image requires the determination of a pixel transfer function that maps pixels from one image to the other. Basically, the

transfer function describes the physical pixel movement (or a 2D flow field) in between these frames.

Computing the image flow field is a thoroughly studied area in computer vision [5]. However, to characterize a 2D flow field resulting from the time recording of a general 3D scene with multiple, independently-moving 3D objects (vehicles and pedestrians) can be very challenging indeed. However, we have found that an elaborate analysis is not needed. Instead, we have developed a highly efficient and accurate global pixel transform model to use in in-painting. Characterizing the pixel flow field by a global motion pattern allows us to use a small number of corresponding unblocked pixels identified in a wiper and an adjacent non-wiper image to ascertain the transformation equation, and apply such a global transform to fill in the wiper pixels in a wiper frame using the corresponding non-wiper pixels in an adjacent non-wiper image. This formulation by-passes the task of finding the right pixels to replace the wiper pixels by a direct color comparison, which is not possible when a wiper pixel has a different color signature from its counterpart in an adjacent non-wiper image.

If we assume that the collective footprint of independently-moving objects (vehicles and pedestrians on the road) is small comparing to that of the stationary background, the dominant pixel flow - induced by the vehicle's ego-motion - is of the pixels on the background. If the background pixels are sufficiently far away, such a flow field can be characterized mathematically by a homography, or an even simpler affine transformation.

Furthermore, most of the time a vehicle's heading should change negligibly in between two adjacent video frames separated by 1/30-th second. So the optical axes of the two neighboring camera shots should align reasonably well. In that case, the pixel movement in-between is well approximated by a zoom. Using a reasonable assumption that the camera's optical axis goes through the center of the image plane (C_x, C_y), then the pixel coordinates in the two adjacent frames should be related by

$$\begin{bmatrix} \alpha_x & 0 & (1-\alpha_x)C_x \\ 0 & \alpha_y & (1-\alpha_y)C_y \\ 0 & 0 & 1 \end{bmatrix}$$

where α's are the unknown zoom factors. A small number of feature correspondences in the two frames (one wiper and the other non-wiper) then can be used to solve for α_x and α_y (in fact, as a first order approximation, α_x and α_y should be the same). Note that this is an affine transform with a special form of only 4 DOFs (α_x, α_y, C_x, and C_y) instead of a general affine with 6 DOFs.

Certainly, the image features used in solving these parameters may come from independently moving objects in the scene. We use the standard RANSAC algorithm [5] to rid the computation of such outliers. That is, we randomly select enough feature correspondences in the wiper and adjacent non-wiper frames to compute (α_x, α_y, C_x, and C_y) and observe how well the computed parameters can explain the feature correspondences of other features. This process is repeated a number of times, and the best zoom parameters from these trials are used.

We smooth over the color difference between a wiper image and its non-wiper neighbor by interpolation. The process is quite straightforward: We first copy over the wiper image to the output (blended) image. We then replace the wiper pixels in the output with the corresponding pixels in the non-wiper neighbor using the estimated pixel zoom model. For a pixel in the output that is not in the wiper region, we compute its distance to the nearest wiper pixel (i.e., a distance transform). For non-wiper pixels in the output that are within a certain blending distance to the wiper region, we smoothly interpolate the colors of the original wiper image with its corresponding non-wiper neighbor (again, based on the zoom pixel motion model). This simple scheme hides the color difference between the two images (a wiper image and its non-wiper neighbor) pretty well.

III. EXPERIMENTAL RESULTS

We have used two data sets, both 720 by 480, for training and validation: a black wiper on gray background (BW) and a red wiper on blue background (RW). Each data set contained three video sequences with wiper speed set at fast, median, and slow. Some sample training images were shown in Figure 2. We had also available five videos taken in real-world driving scenarios for testing real-world deployment.

Individual image frames were extracted from these videos using public-domain programs (ffmpeg and virtualdub). For BW, 517 sample wiper frames from among the first 3,000 frames in the fast sequence were manually selected for training. The K-mean segmentation results of 21 out of the 517 selected frames were visually poor, and hence, these 21 frames were excluded from the process of building the eigen wipers. This corresponds to a failure rate of 4%.The same procedure was used for the RW data set and 425 wiper frames were extracted from the fast sequence. Of these 425 frames, 74 had bad segmentation results by a visual inspection, or a 17% failure rate.

In building the eigen space, we have used an area threshold of 10%, i.e., the detected wiper region must contain at least 10% of the image pixels. Furthermore, we have used an accuracy threshold of 1%, or the singular value of a eigen wiper must be at least 1% of the average of the top 3 singular values to be deemed significant and used for model building. For the BW data set, 465 out of 517 images survived the area thresholding and 330 of the 465 remained were deemed significant and used for building the eigen space. For the RW data set, 344 out of 425 images survived the area thresholding and 297 of the 344 remained were used for building the eigen space

In the validation stage, we manually selected from all three BW sequences both positive and negative samples (for the fast sequence, we made sure that the test images were from frames 3,001 onward with no overlap with the training images). For RW sequence, as the training set exhausted all wiper images from the fast sequence, we used only the median and

slow sequences for validation. We have used the Euclidean distance in our computation.

Some preliminary results are summarized in the following tables. Table 1 shows the histograms of the second matching score of the three video sequences in the BW data set. Histograms of the second matching scores of both the labeled wiper images (red) and labeled non-wiper images (blue) are shown. The best threshold to separate the wiper from the non-wiper images are listed below the plots. In Table 1, we summarize the misclassification rates of the three sequences for wiper images, non-wiper images, and their averages. The corresponding results of the RW data set are shown in Table 2. The results show excellent separation of wiper and non-wiper classes using our eigen representation.

The processing flow of in-painting is to first classify a video frame as either a wiper or a non-wiper frame. For a wiper frame, we search for the closest non-wiper frame either before or after in the onboard video (as shown in Figure 3, such a neighboring non-wiper frame is always found within a few frames away). We detect and match SIFT features in these two frames to identify a number of feature correspondences (i.e., 2D points resulting from the projection of the same 3D point). We then solve the zoom pixel-flow equation using these correspondences. Applying the pixel mapping equation thus obtained, we can map unblocked pixels from the non-wiper frame to replace the corresponding wiper pixels in the wiper frame.

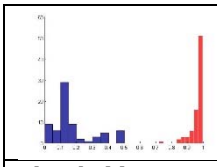
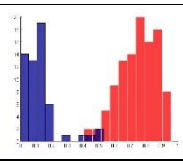
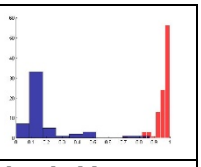
Figure 4 presents a few examples where the left figures show localized wiper region in red (with the corresponding pixels used in computing the zoom model marked in some of these examples). The right figures show the in-painting results (with pixels in the red wiper region replaced by the corresponding unblocked pixels in the adjacent non-wiper frame).

It is important to note that the program can choose a non-wiper frame either before or after the wiper frame for in-painting. Ideally, the neighboring in-painting photo should be BEFORE the wiper photo in the video to avoid seeing red blocks around the image periphery. Refer to Figure 5, if the in-painting neighbor is AFTER the wiper image, the vehicle would have moved forward. Hence, the field of view (FoV) of the neighboring in-painting frame is smaller than that of the wiper frame. Therefore, wiper pixels around the periphery of the wiper frames have no counterpart in the neighboring in-painting frame and cannot be replaced. This behavior is mathematically correct and unavoidable. However, as the red pixels are around the frame boundaries as shown in Figure 5, the visual distraction is hopefully small.

IV. CONCLUDING REMARKS

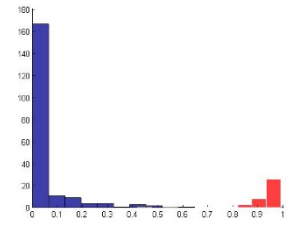
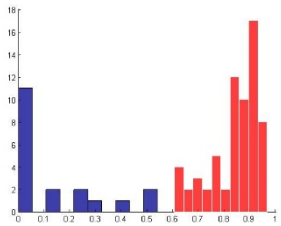
As can be seen from the experimental results, one major deficiency of our current implementation is in wiper extraction. If a wiper is not completely and cleanly extracted from a wiper frame, darkened pixels around the wiper's periphery remained in the blended images and were quite noticeable. Improving wiper segmentation is therefore a topic to work on in the future.

Table 1 BW training set results, wiper images in red and non-wiper images in blue and the second table shows the misclassification rate

		
Threshold: 0.6	Threshold: 0.3	Threshold: 0.6
Fast	Median	Slow

Validation data set	Fast	Median	Slow
Size	127 wipers, 70 non-wipers	167 wipers, 57 non-wipers	145 wipers, 53 non-wipers
Labeled wiper images	0%	0%	0%
Labeled non-wiper images	0%	7%	4%
Average misclassification rate	0%	2%	1%

Table 2 RW training set results, wiper images in red and non-wiper images in blue, the second table shows the misclassification rate

	
Threshold: 0.5	Threshold: 0.6
median	slow

Validation data set	Median	Slow
Size	64 wipers, 201 non-wipers	97 wipers, 19 non-wipers
Labeled wiper images	2%	0%
Labeled non-wiper images	0%	4%
Average misclassification rate	1%	0%

Figure 4 Sample results of wiper region detection and localization (left) and in-painting (right)

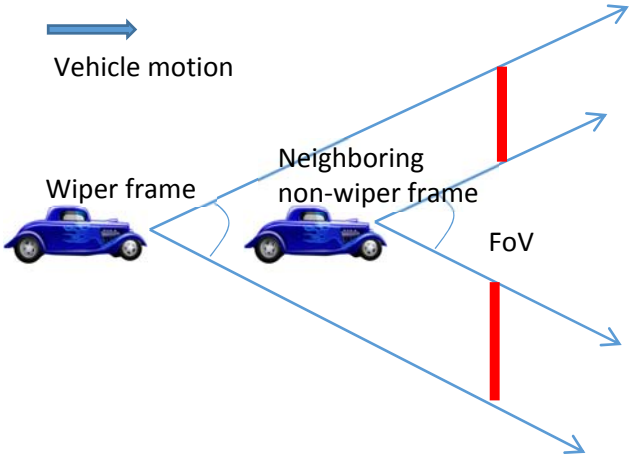


Figure 5 Cause of red border in in-painting images

REFERENCES

1. G. Welch and G. Bishiop, http://www.cs.unc.edu/~welch/kalman/
2. Cappe, O.; Godsill, S.; Moulines, E.; "An Overview of Existing Methods and Recent Advances in Sequential Monte Carlo". *Proceedings of IEEE* **95** (5): 899-924, 2007.
3. Hastie, Tibshirani, and Friedman, *the Elements of Statistical Learning*, Springer, 2001.
4. L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces". *Journal of the Optical Society of America A* **4**(3): 519–524, 1987.
5. R. Szeliski, *Computer Vision: Algorithms and Applications*, Springer-Verlag, 2010.
6. R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, MA, 2003.