ROBUST AND EFFICIENT TRACKING WITH LARGE LENS DISTORTION FOR VEHICULAR TECHNOLOGY APPLICATIONS

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

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

Abstract— Advances in video technology have enabled its wide adoption in the auto industry. Today, many vehicles are equipped with backup, front-looking, and side-looking cameras that allow the driver to easily monitor traffic around the vehicle for enhanced safety. One difficulty with performing automated image analysis using a vehicle's onboard video has to do with the significant lens distortion of these sensors to cover a large field of view around the vehicle. This paper reports our research on proposing a tracking scheme that improves the accuracy and denseness of object tracking in the presence of large lens distortion. The contribution of our research is 4-fold: (1) We evaluated a large collection of state-of-the-art trackers to understand their deficiency when applied to videos with large lens distortion, (2) we showed how to derive useful evaluation metrics from public-domain, real-world driving videos that do not come with ground-truth information on pixel tracking, (3) we identified many enhancement techniques that can potentially help improve the poor performance of current trackers on videos of large lens distortion, and (4) we performed a systematic study to validate the efficacy of these enhancement techniques and proposed a new tracker design that achieved substantial improvement over the state-of-the-art, in terms of both accuracy and density, based on a rigorous prevision vs. recall analysis.

I. INTRODUCTION

Video cameras are becoming ubiquitous in the modern societies. They are increasingly being adopted by the auto industry for its falling price and improving capabilities. The U.S. National Highway Traffic Safety Administration (NHTSA) issued a rule in 2014 requiring all new light vehicles sold or leased in the U.S. to have "rear-view visibility systems," in effect, requiring backup cameras. The rule would start phasing in on May 1, 2016 models and be at 100% May 1, 2018. Under the rule, all vehicles would have to give the driver a view 10-foot by 20-foot zone behind the vehicle. There are also requirements for image size and other factors that all but require rear-view cameras as the only solution.

However, availability of such a system does not fully prevent back-over accidents from happening. Many times, a driver can be distracted by, say, passengers in the vehicle and events surrounding the vehicle, and fails to properly observe the video display. An added safety measure is to have an automated monitoring system to track and identify obstacles in the video and sound an alarm if an obstacle comes into close vicinity for potential collision. This is similar to other vehicular safety systems, such as the lane departure warning system and the adaptive cruise control system, that perform environmental monitoring automatically without human intervention. The goal is thus to study issues in designing such an automated image analysis system for use with a rear-view camera.

II. TECHNICAL RATIONALE

The added degree of difficulty for back-over detection is that to observe obstacles (vehicles, pedestrians, animals, fences, gates, mailboxes, shopping carts, vegetation, etc.) that a vehicle might come into contact with in a reverse motion, the view of the camera must be wide enough to observe not just what is directly behind the vehicle, but also what is to the side that can potentially moves into the vehicle's path. This necessitates the use of a lens with large spherical distortion (e.g., a fish-eye lens). In Figure 1, we show a sample frame recorded by such a backup camera (top left) and the distortion corrected frame (top right). We also depict how pixels in the original, distorted image move during the distortion correction process as colored flow vectors overlapped on the original, distorted (bottom left) and distortion corrected (bottom right) images.

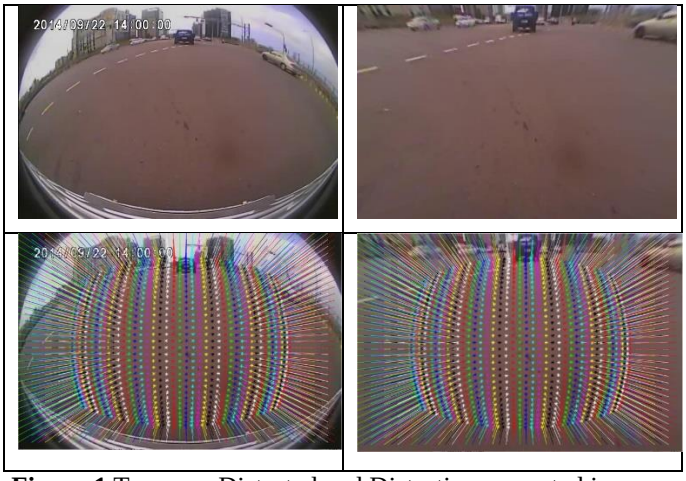

Figure 1 Top row: Distorted and Distortion corrected images and bottom row: with pixel movement (from distorted image to distortion-corrected image) overlaid**.**

The choice of a back-over warning system seems to be either performing image analysis directly on the original, distorted images or correcting for lens distortion before attempting image analysis. The former often implies a significant redesign of the image analysis algorithms. This is

because a back-over warning system often implements a tracking scheme to detect and track moving obstacles, and state-of-the-art object trackers perform sophisticated feature detection and matching to guard against scale, lighting and viewpoint changes. For example, SIFT [2] and SURF [3] build hierarchical image pyramids; perform pyramidal, scale-space analysis to locate features; and describe features using histograms of local gradient patterns. The smoothing operations, histogram binning and scale-space computation all assume a regular, Cartesian grid. It is highly non-trivial to reformate the framework using a non-linear grid.

Figure 2 Sample results of applying standard trackers to distortion corrected images. First row: two input frames for tracking, second row: accurate feature-based trackers (left: SIFT, right: SURF), third row: efficient feature-based trackers (left: BRIEF, right: ZBRIEF), and last row: flow-based trackers: (left: KLT, right: dense optical flow).

As is evident from Figure 1, the lens distortion is non-uniform and significant as pixel movements can be as large as 20% or more of the image size around image periphery. Furthermore, as a small region in the center of the distorted image is (nonlinearly) expanded to fill in the whole image frame in the distortion corrected image, original pixels must often be used multiple, unequal numbers of time in a pyramidal smoothing scheme. The expansion of the central image region then introduces artifacts that are not in the original images, and distinct defects may be evidenced using different interpolation schemes. Furthermore, to fit lens with different distortion parameters, such a scheme might have to be re-worked and re-optimized multiple times.

Hence, we advocate the latter approach where lens distortion is corrected first and then a standard tracking scheme is applied on the corrected images. Conceptually, this is a simpler approach without redesigning trackers, but it faces the same difficulty that nonlinear stretching of the original image introduces artifacts and noises. To illustrate, Figure 2 shows six different trackers in three rows. These include both feature-based (2nd and 3rd rows) and flow-based (4th row) trackers, and trackers emphasizing accuracy with multiple mechanisms to guard against incidental environmental variation (as in SIFT [2], SURF [3] and optical flow [1,8]) and trackers emphasizing efficiency that trades off some safeguard for speed (as in BRIEF [5], ZBRIEF [13] and KLT [9,10]). However, most of them produced sparse correspondences. The sparsity is especially noticeable for, say, homogeneous, featureless pavements, where features are noisy and unstable in distortion corrected images that defeat tracking. Note that though dense optical-flow based approaches generate dense correspondences, the computed flow does not depict an expected zoom flow and is visually incorrect.

While one may tune the tracker parameters to increase feature density, densification invariably involves a trade-off between precision and recall. Intuitively speaking, given, say, a pair of images, the amount of information embedded therein is fixed. A feature detector, with its particular algorithmic design, has an inherent limit on what it can accurately detect and describe. That is, the trade-off between precision and recall dictates that when the "floodgate" is open wider, more pairings will be generated, but with less assurance of the quality of the matches. This is illustrated in Figure 3 that a lower precision setting does admit more pairings. But the quality is much poorer. Hence, it is not sufficient to just "let more in;" we need to exercise caution to ensure that matching quality does not degrade significantly as a result.

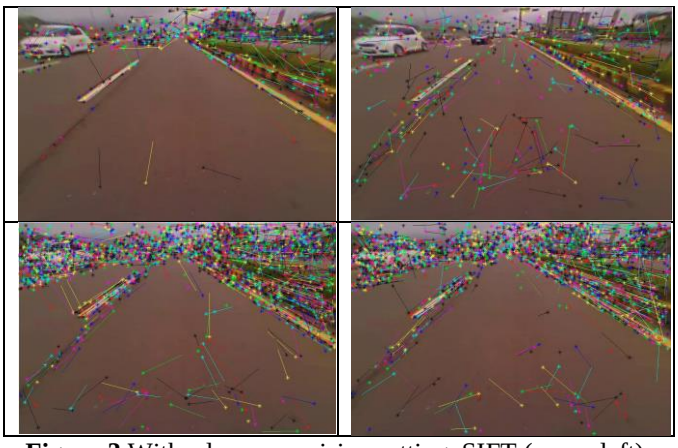

Figure 3 With a lower precision setting, SIFT (upper left), SURF(upper right), BRIEF (lower left) and ZBRIEF (lower right) all admit more feature pairing, but with poorer quality.

As tracking is a fundamental operation in video analysis and collision avoidance, it is paramount that the performance of these trackers be significantly improved on such videos of large lens distortion. In this paper, we conducted a study of a number of enhancement mechanisms to verify their efficacy in such an endeavor. We show that by properly combining these enhancement techniques, we can improve both the accuracy and density of the tracking results.

A. Evaluation Metrics

Note that while the distortion compensated videos appear visually correct (e.g., lines are straight instead of curved), there is no ground truth associated with these data sets. That is, no correct feature correspondences are available. Certainly, some feature movement patterns can be predicted on such driving videos when the vehicle's trajectory is known. It is well known that corresponding features in two views are related by a simple epipolar constraint, expressed mathematically as $\mathbf{x'}^{\mathrm{T}}\mathbf{F}\mathbf{x}=0$, where **F** is the Fundamental Matrix [11] and **x** and **x'** are the corresponding feature coordinates in two frames. It was shown [11] that in the case of a pure translation (a vehicle is moving straight ahead or back), **F** has a very simple expression $\mathbf{F} = [\mathbf{e'}]_x$. where **e'** is the epipole and $[]_x$ is the anti-symmetric 3 by 3 matrix of the enclosed vector.

Two typical cases during driving are illustrated here: If the camera is mounted on the left or right side of the vehicle, most of the times the camera will be translating horizontally and $\mathbf{e'} = (1, 0, 0)^{\mathrm{T}}$ and the epipolar line is $y' = y$ (or pixel movement should be purely horizontal). When the camera is translating along the **z** direction (for a front- or back-mounted camera), $\mathbf{e'} = (c_x, c_y, 1)^{\mathrm{T}}$, where (c_x, c_y) is the focus of expansion, or

$$\mathbf{F} = \begin{bmatrix} 0 & -1 & c_y \\ 1 & 0 & -c_x \\ -c_y & c_x & 0 \end{bmatrix} \qquad \mathbf{1}$$

The epipolar line in the 2nd (primed) image is therefore:

$$f(x', y') = -(y - c_y)x' + (x - c_x)y' - c_y x + c_x y = 0$$

This is the equation of a line going through (c_x, c_y) $(f(c_x, c_y) = 0)$ with a direction $(x - c_x, y - c_y)$, which represents either a convergent (zoom-in or front-mounted) or divergent (zoom-out or back-mounted) flow pattern.

However, such a prediction does not provide unique pixel-to-pixel correspondences between features in two frames. Lacking knowledge of the 3D scene structure, such a mathematical model only predicts the direction of the pixel flow, and hence, only the directional error, not the absolute error (direction and magnitude), can be ascertained. However, barring other sources of information, we will use directional error and feature pairing density as metrics to evaluate different algorithmic enhancements.

B. Potential Remedies

Here we list a number of possible remedies that we have experimented with in the course of our research.

Pyramidal processing: For a rear-view camera, magnitude of pixel movement is usually not constant and gets progressively larger closer to the vehicle. The search neighborhood for feature pairs should be enlarged to accommodate potentially significant pixel movements or images should be sub-sampled to keep the search neighborhood a reasonable size. Standard sub-sampling method is to build a Gaussian pyramid for coarse-to-fine processing,

Image smoothing: Irregular and inconsistent pairing results shown in Figure 1 and Figure 2 often indicate noisy image content. This is especially true for featureless regions (e.g., pavement) where random perturbation of color and intensity due to environmental factors and distortion correction can be overwhelming. Standard practice is to smooth out such undulation,

Median filtering: As indicated by recent research [8], localized median filtering is a powerful tool to rid the flow field of outliers and hence, is worthy of experimenting, and

Regularization: As mentioned before, while epipolar constraint is not strong enough to ascertain unique pixel-to-pixel correspondences, it does provide motion-specific restriction on the direction of the pixel flow. Such domain-specific constraint can be used as a regularization factor, and

Magnitude filtering: One drawback of KLT, BRIEF, and ZBRIEF is that, for efficiency consideration, features are often located on the integer grid. Without sub-pixel precision, quantization error in localization can be as large as half a pixel. Such a quantization error is most pronounced when the magnitude of the pixel movement is small, as can be seen in the following equation:

$$\tan\hat{\theta} = \frac{\hat{v}_y}{\hat{v}_x} = \frac{v_y + \delta v_y}{v_x + \delta v_x} = \frac{v_y(1 + \frac{\delta v_y}{v_y})}{v_x(1 + \frac{\delta v_x}{v_x})} \cong \frac{v_y}{v_x}\left(1 - \frac{\delta v_x}{v_x} + \frac{\delta v_y}{v_y}\right) = \tan\theta(1 - \frac{\delta v_x}{v_x} + \frac{\delta v_y}{v_y}) \qquad 2$$

Where the "hat" quantities are defined on an integer grid and the error is proportional to $\left|\frac{\delta v_x}{v_x}\right| + \left|\frac{\delta v_y}{v_y}\right|$. Hence, to reduce the quantization effect, we can ignore movement that is small than a certain threshold.

Of all these schemes, we have found that pyramidal processing and image smoothing to be relatively ineffective. First, pyramidal processing is already employed in flow-based schemes (KLT and optical flow) and in feature-based schemes (SIFT and SURF), but did not produce satisfactory results as shown in Figure 1 and Figure 2. Second, we have experimented with many different interpolation schemes, from simple box filter, to more advanced Gaussian and spline filters in building pyramids and smoothing images, but none of them produced good tracking results. We surmise that the large nonlinear stretching of pixels in distortion correction inevitably introduces artifacts as few seen pixels are used to generate significantly more unseen pixels among them. This theoretical limitation of lack of information cannot be overcome by clever interpolation. Hence, we will focus on the other enhancement schemes in the next section.

III. Experimental Results

We used four cameras with wide-angle lens and mounted them on the front, rear, left and right sides of a test vehicle to simulate

the possibility of collecting videos with large lens distortion for monitoring traffic all around a vehicle (Figure 4), not necessarily just that coming up from the rear of the vehicle. We collected about 100 frames from each of these four videos (400 frames total) as our test bed. We ran SIFT, SURF, USURF (a variant of SURF that does not handle in-plane rotation), BRIEF, ZBRIEF (a variant of BRIEF that handles zoom motion typical in stiraight line forward and backup driving) and KLT on them.

Figure 4 Sample frames from front (top left), back (top right), left (bottom left) and right (bottom right) sequences.

We have used two metrics for evaluation: directional error (precision) and feature density (recall). To study the trade-off between precision and recall, we varies a single parameter that controls the "flood-gate" of the number of admitted feature pairs. For feature-based trackers, they all employ a descriptor that records the local intensity profile, SIFT and (U)SURF using localized gradient histograms and (Z)BRIEF using a localized random sampling pattern. SIFT's descriptor is a vector of length 128, (U)SURF a vector of length 64, and (Z)BRIEF's descriptor can be 64, 128, or 256 bits long. Feature similarity is computed as the Euclidean distance between two SIFT and (U)SURF descriptors or the Hamming distance between two (Z)BRIEF descriptors.

Our correspondence scheme requires matched feature pairs to share similar descriptors (small Euclidean or Hamming distance) and be unique (the best match should beat out the second best match by some preset margin). If this margin is set to, say, 0.7, this means that the feature distance of the best match must still be smaller than that of the second best, even if the second best is decreased by a factor of 0.7. Obviously, smaller (tighter) margins imply more unique matches (that the best match still stands out even if the errors of other potential matches are artificially lowered) and vice versa. Hence, to generate more matches, we simply relax or increase this margin (we have tried six different values, 0.7, 0.75, 0.8, 0.85, 0.9 and 0.95 in our experiments). Then, for each parameter setting of each enhancement scheme, we record the average number of feature pairs obtained on the test sequences and the degree of discrepancy of these pairings to the appropriate epipolar constraints (either a pure translational motion for left and right mounted camera or a zoom in and out motion for a front and back mounted camera).

For a flow-based tracker, KLT employs bi-directional tracking error as a quality metric. For a pair of frames, feature tracking can be initiated from either frame. If a feature is tracked from one frame to the other and back, discrepancy between the starting and end locations is used to measure consistency. Allowing large discrepancy admits more features.

A. Baseline Study

Figure 5 presents the baseline study on four cases: SIFT, SURF, USURF, and ZBRIEF. The baseline cases do not employ any enhancement mechanism, i.e., no magnitude thresholding, no regularization, and no median filtering is applied. The average precision (directional error) is shown in *x* and the average recall (number of feature correspondences per image pair) is shown in *y*. For each tracker, the matching margin varies from the tightest (0.7, marked as 'o') to the loosest (0.95, marked as '+'), with intermediate ones marked in '*' and strung in a curve. Note that we have chosen to present the ZBRIEF results as ZRBIEF detector is designed for detecting a zoom in or zoom out flow, and hence, is well suited for a rear-view camera that is central to this study. Figure 6 shows sample tracking results graphically where the left column has the tightest margin (0.7) and the right column has the loosest margin (0.95) and from top to bottom are SIFT, SURF, USURF, and ZBRIEF.

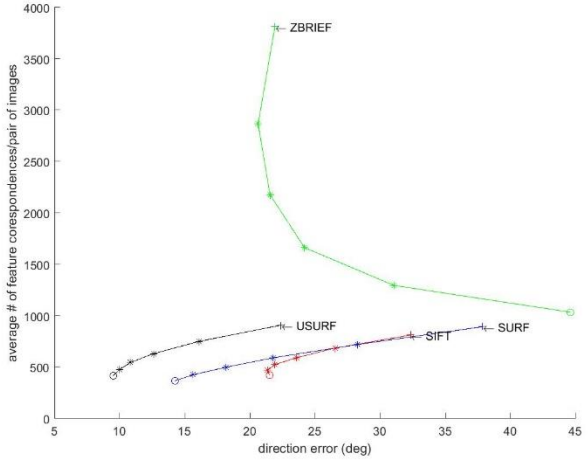

Figure 5 Baseline study where no enhancement mechanism is used. Four curves SIFT (red), SURF (blue), USURF (black), and ZBRIEF (green) are shown. For each curve, the tightest pairing margin (0.7) is marked as a circle (o), the loosest margin (0.95) marked as a plus (+), and intermediate margins are strung in a curve and marked as stars (*).

A number of observations can be made on the results: (1) As the pairing margin is relaxed, the flow becomes denser (from left to right in Figure 6), (2) ZBRIEF (bottom row) produces much denser results than SIFT (top row), SURF (2nd top) and USURF (3rd top), (3) feature density on the homogeneous pavement is quite low, almost non-existent for SIFT at all matching margin levels. (U)SURF and ZBRIEF provide higher flow density on pavements, but the accuracy is visually poor, (4) the precision vs. recall trade-off holds for SIFT, SURF and USURF. As seen in Figure 5 these curves of relaxing margin grow from lower left to upper right, implying

larger angular errors allow higher flow densities, and (5) it is slightly puzzling that the ZBRIEF curve (green) does not follow the trend – the directional error actually drops as the density increases. This abnormality could be explained by observing that, using a tighter threshold, few ZBRIEF features are matched resulting in a large flow. However, small flow vectors have much larger directional errors as shown in Eq. 2 [13]. With more relaxed thresholds, larger flow vectors are detected which help with reducing the directional error.

Figure 6 Sample baseline tracking results. Left column: the paring margin set at 0.7 (tightest) and right column: the paring margin set at 0.95 (loosest). From top to bottom: SIFT, SURF, USURF and ZBRIEF.

B. Baseline + Regularization

In order to understand how the three potential enhancement schemes help, we apply them independently, one at a time. We use four regularization factors, four median filter setting, and four magnitude thresholds. An exhaustive search would result in 64 possible combinations. Instead, we take a greedy approach. We apply these schemes sequentially. For each scheme used, we identify the best setting and then use that setting for the next scheme.

The component that proves most beneficial, from our experience, is the *in vitro* regularization. Regularization is a well-established technique in computer vision [12]. In addition to appearance similarity and pairing uniqueness, we incorporate regularization using a penalty term. We add a positive penalty to the distance measure so as to decrease the likelihood of the pairing being used if the flow direction is wrong. This penalty is proportional to the misalignment in angles between the computed flow and the theoretical epipolar direction. Hence, the regularization factor is used in the tracker itself during the feature pairing phase, instead of being applied after-effect. Furthermore, using a penalty expression allows us to impose regularization on individual matches efficiently.

Figure 7 shows the results using regularization on the four feature-based trackers. As seen in Figure 7, regularization helps to significantly reduce the directional errors in all four cases. Furthermore, the pairing densities maintain at roughly the same level even with the largest regularization factor (black lines in Figure 7). That is, the red curves (baseline) stays to the right of the green, blue and black curves (baseline + regularization), which indicates reduced directional errors. Furthermore, all these curves have roughly the same height, indicating approximately the same density at the corresponding parametric settings. The difference among regularization factors is actually not significant. We have used the factor of 2 for SIFT and (U)SURF and 120 for ZBRIEF (the blue lines in Figure 7) in all later studies**.**

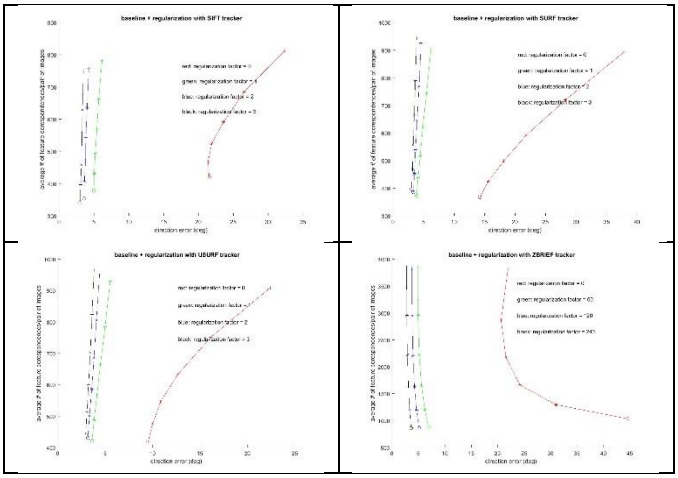

Figure 7 Baseline + regularization study where four different regularization levels are used. Each subgraph contains four curves, with varying regularization factors (red, green, blue, and black). For each curve, the tightest margin (0.7) is marked as a circle (o), the loosest margin (0.95) marked as a plus (+), and intermediate margin values are strung in a curve and marked as stars (*). Top left: SIFT, top right: SURF, bottom left: USURF, bottom right: ZBRIEF.

C. Baseline + Regularization + Median Filtering

We perform median filtering by playing with the neighborhood size (from 20, 40 to 60 pixels across) and maximum number of neighbors used (from 5, 7 to 9) in the process. As seen in Figure 8, the general trend is that median filter helps to reduce the directional errors in all trackers while maintaining the pairing densities at roughly the same level. That is, the red curves

(baseline + regularization) stay to the right of the green, blue and black curves (baseline + regularization + median filtering), which indicates reduced directional errors. Furthermore, all these curves have roughly the same height, indicating approximately the same density at the corresponding parametric settings. The difference among median filters is not significant. We have used a median filter of size 60x60 and a maximum neighbors of 9 (the black curves in) in all later studies.

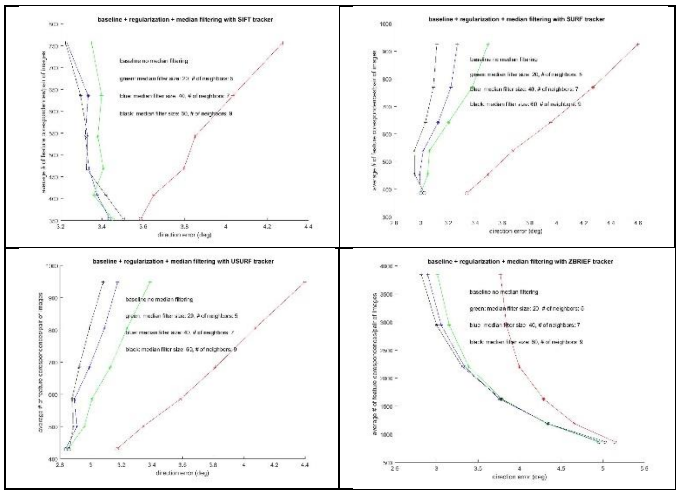

Figure 8 Baseline + regularization + median filtering study where four different median filters were used. Each subgraph contains four curves, with median filter set to smallest (20x20 and 5 neighbors most) to largest (60x60 with 9 neighbors most) in red, green, blue to black. For each curve, the tightest margin (0.7) is marked as a circle (o), the loosest margin (0.95) marked as a plus (+), and intermediate margin values are strung in a curve and marked as stars (*).

D. Baseline + Regularization + Median Filtering + Magnitude Threshold

A final mechanism is to threshold the flow magnitude and remove small flow vectors from consideration. The results are summarized in Figure 9. As seen in Figure 9, eliminating small vectors helps the accuracy, but the density drops as a result. We have settled on using a threshold value of 3.

Using these enhancement mechanisms and their optimal values as identified, we process the same images shown in Figure 6 and depict the results graphically in Figure 10. The results shown in these two figures are in correspondence and the difference is that Figure 6 does not have any enhancement scheme applied while Figure 10 have all enhancements. The improvement from Figure 6 to Figure 10 should be apparent to the naked eyes, demonstrating the effectiveness of the combination of the enhancement schemes for all trackers.

The space limitation does not allow us to give an in-depth discussion of these mechanisms on flow-based trackers. Dense optical flow frameworks, such as that proposed in [8], already employ most of these mechanisms. Hence, it is puzzling why the resutlts are poor as shown in Figure 2. We do use these mechanisms with the KLT trackers and observe similar improvement as with the feature-based trackers.

Finally, we address the issues of efficiency. None of these mechanisms – regularization, median filtering, and magnitude filtering – proves to be computationally expensive. Regularization can be an expensive proposition, if it is formulated as a global optimization framework [12]. We avoid such an overhead by proposing only a penalty term, which can be computed efficiently. Many of these mechanisms represent just a few extra lines in the tractor codes and take negligible time comparing to all other steps combined. Our estimate is that they add about 3% to 5% of the total run time, as the majority of the processing, e.g., for SIFT and SURF in pyramidal, scale-space analysis, is done elsewhere.

Based on the results, we recommend the use of ZBRIEF, as it produces results that have similar accuracy as SIFT and (U)SURF but with a feature density that is 4 times higher (Figure 9). Furthermore, we show that ZBRIEF is 5 to 6 times faster than SIFT and SURF in our other analysis [13]. Some more examples of using ZBRIEF on videos collected with large lens distortion are shown in Figure 11. Finally, large stretching in lens distortion correction might have resulted in images with a low signal-to-noise ratio. An elaborate scheme like SIFT may describe the noise instead of the intrinsic pattern, that might explain its poor performance.

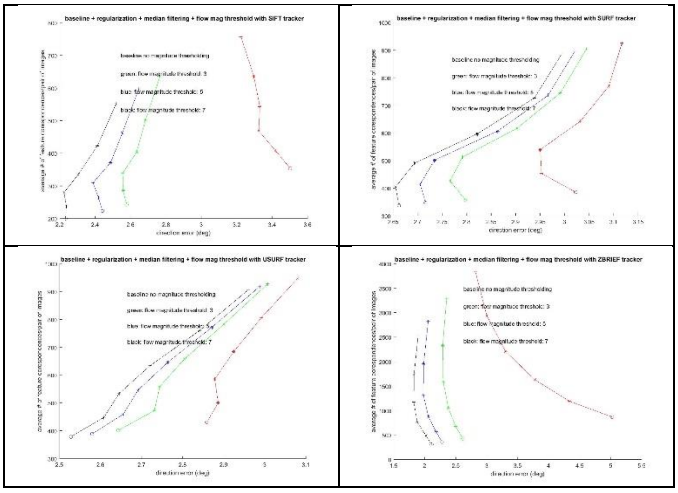

Figure 9 Baseline + regularization + median filtering + flow magnitude threshold study where four different magnitude threshold were used. Each subgraph contains four curves, with magnitude threshold set to 0, 3, 5, 7 pixels in red, green, blue to black. For each curve, the tightest margin (0.7) is marked as a circle (o), the loosest margin (0.95) marked as a plus (+), and intermediate margin values are strung in a curve and marked as stars (*).

IV. CONCLUDING REMARKS

This paper summarizes our research on feature tracking in the presence of large lens distortion for vehicular technology applications. We have conducted an extensive evaluation of many state-of-the-art trackers, identified their deficiency when applied to videos with large lens distortion, and proposed enhancement schemes for improvement based on a rigorous precision vs. recall analysis. Certainly, a robust back-over

warning system may encompass many components; such an efficient and robust tracker should hopefully prove useful.

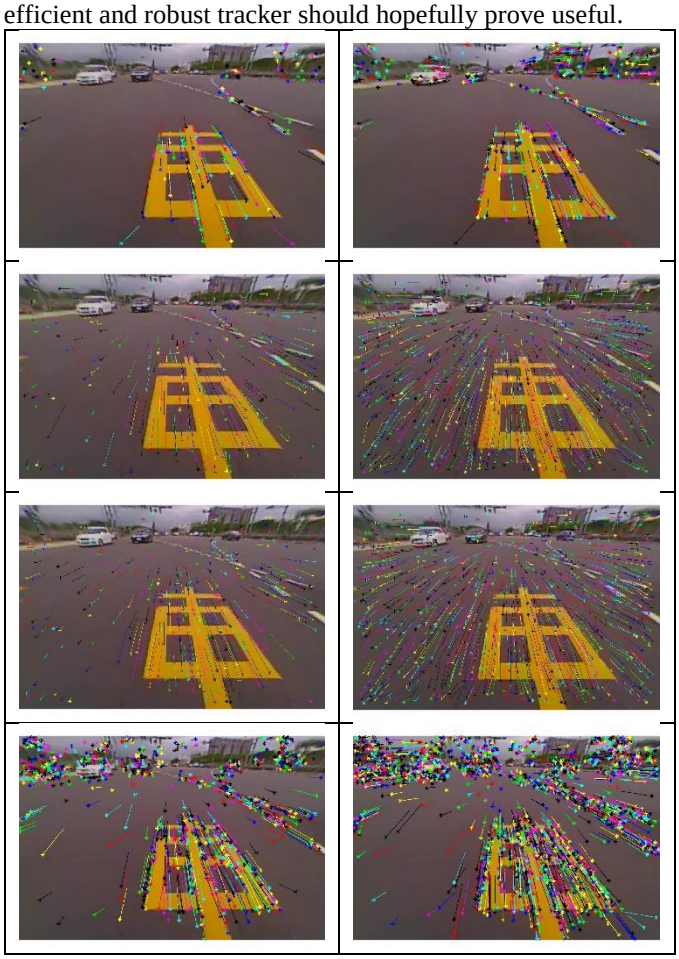

Figure 10 Sample baseline + regularization + median filter + magnitude threshold tracking results. Left column: the paring margin set at 0.7 (tightest) and right column: the paring margin set at 0.95 (loosest). From top to bottom: SIFT, SURF, USURF and ZBRIEF. These figures are in one-to-one correspondence with those in Figure 6 to show the enhancement using our analysis.

REFERENCES

1. Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael Black, Richard Szeliski, A Database and Evaluation Methodology for Optical Flow, International Journal of Computer Vision, 92(1):1-31, March 2011.
2. Lowe, D. G., “Distinctive Image Features from Scale-Invariant Keypoints”, International Journal of Computer Vision, 60, 2, pp. 91-110, 2004.
3. Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool, "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346-359, 200.
4. E. Rosten and T. Drummond, "Machine learning for high-speed corner detection". European Conference on Computer Vision. Springer. pp. 430–443, 2006.
5. M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua, “{BRIEF}: Computing a Local Binary Descriptor Very Fast,” IEEE Transactions on Pattern Analysis and Machine Intelligence 2012.
6. Mikolajczyk, K., and Schmid, C., "A Performance Evaluation of Local Descriptors", IEEE Transactions on Pattern Analysis and Machine Intelligence, pp 1615--1630, 2005.
7. Ethan Rublee Vincent Rabaud Kurt Konolige and Gary Bradski, “ORB: an efficient alternative to SIFT or SURF”, International Conference on Computer Vision, 2011.
8. Sun, D.; Roth, S. & Black, M. J. "Secrets of Optical Flow Estimation and Their Principles" IEEE Int. Conf. on Comp. Vision & Pattern Recognition, 2010.
9. Bruce D. Lucas and Takeo Kanade, “An Iterative Image Registration Technique with an Application to Stereo Vision,” *International Joint Conference on Artificial Intelligence*, pp. 674–679, 1981.
10. Jianbo Shi and Carlo Tomasi, “Good Features to Track,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600, 1994.
11. R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, MA, 2003.
12. B. K. P. Horn, *Robot Vision*, MIT Press, Cambridge, MA, 1986.
13. Che-Tsung Lin, Long-Tai Chen, and Yuan-Fang Wang, "Evaluation, Design and Application of Object Tracking Technologies for Vehicular Technology Applications ," *Proceedings of IEEE Vehicular Technology Conference*, Boston, MA, 2015.

Figure 11 More sample results using ZBRIEF. Left: with a tight margin and without any enhancement and right: with a loose margin and with all enhancements. First two rows are from a front-mounted camera with a zoom-out flow, 3rd row is from a back-mounted camera with a zoom-in flow and last row is from a side-mounted camera for a translational flow.