2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Light-weight Localization for Vehicles using Road Markings

Ananth Ranganathan and David Ilstrup
{aranganathan, dilstrup}@honda-ri.com
Honda Research Institute USA Inc

Tao Wu
taowu@umd.edu
University of Maryland

***Abstract*—Traditional vision-based localization methods such as visual SLAM suffer from practical problems in outdoor environments such as unstable feature detection and inability to perform location recognition under lighting, perspective, weather and appearance change. Additionally map construction on a large scale in these systems presents its own challenges. In this work, we present a novel method for precisely localizing vehicles on the road using signs marked on the road (road markings), which have the advantage of being distinct and easy to detect, their detection being robust under changes in lighting and weather. Our method uses corners detected on road markings to perform localization in global coordinates. The method consists of two phases - a mapping phase when a high-quality GPS device is used to automatically survey road marks and add them to a light-weight "map" or database, and a localization phase where road mark detection and look-up in the map, combined with visual odometry, produces precise localization. We present experiments using a real-time implementation operating in a car that demonstrates the improved localization robustness and accuracy of our system even when using road marks alone. However, in this case the trajectory between road marks has to be filled-in by visual odometry, which contributes drift. Hence, we also present a mechanism for combining road-mark-based maps with sparse feature-based maps that results in greater accuracy still. We see our use of road marks as a significant step in the general trend of using higher-level features for improved localization performance irrespective of environment conditions.**

I. INTRODUCTION

Precise localization is a pre-requisite in vehicles for tasks ranging from driver assistance to autonomous driving. In this respect, lane-level localization with an error of a few meters is the desired goal in most cases. However, use of GPS alone cannot guarantee this level of accuracy in all environments unless expensive additions such as Differential GPS (DGPS) and high-grade IMUs are used [18][28]. The most reliable solution to the vehicle localization problem thus far has been through the use of 3D lidar sensors, such as the Velodyne [10], which in addition to being expensive, are active sensors with their own set of challenges.

Cameras provide a cheap and attractive sensing alternative to 3D lidars. A popular means of localizing using cameras is visual simultaneous localization and mapping (VSLAM) [1], [11]. In its most basic form, VSLAM consists of building a map by detecting features in the environment. The features used are usually points, resulting in a sparse feature-based point cloud representation for the map. While mapping and localization can be done simultaneously (SLAM), it is more common for the map that is built to be stored for later use [23]. This is because running SLAM in real-time in large scale environments can be challenging whereas localizing against a pre-built map is much less computationally intensive. When localizing against a previously built map (or when closing loops while building maps), the currently detected features are matched to those in the map to find the approximate current location. This is followed by a refinement of the pose estimation using geometric matching.

While a large body of research on VSLAM exists, many practical problems remain. Although self-consistent maps can be built with state-of-the-art algorithms [7], building large scale maps that adhere precisely to some global coordinate frame is problematic [26]. However, for vehicles operating on the road, global coordinates are more suitable than self-consistent maps in an arbitrary coordinate system. Additionally, the basis step of feature extraction, on which the whole VSLAM pipeline relies, can be unstable. This is especially true with changing lighting and appearance conditions [29]. In such cases, it is hard to localize against a pre-built map since feature matching fails. Hence, the need is for higher-level features that are stable against environment changes but may be specific to certain usage scenarios only rather than being completely general.

The main contribution of this paper is a method to use higher level features, in the form of road markings, towards solving the problem of robustness in localization using cameras. We present a method for localizing using corners detected on road markings. Road marks, such as arrows and speed limits, are distinct, relatively easily detectable, and fairly frequent on roadways. Since they are high contrast objects, their detection, and the detection of corners within them, is less susceptible to lighting changes than general point features such as Harris corners. We use our previous work on road marking detection [30] to detect and recognize the type of road markings.

Our system consists of two phases. In the mapping phase, a high-grade GPS+IMU mounted in a car equipped with a camera is used to detect road marks and compute GPS locations of the corners of interest within them. Our "map", which simply consists of road mark labels and GPS coordinates of corresponding corners, is thus extremely light-weight and can scale up to large areas easily. During the localization phase, road marks are recognized, corners within them are detected, and looked up in the map. Knowing the GPS locations of the corners in the current image allows the instantaneous computation of the camera's GPS location also. We combine these instantaneous pose estimates with visual odometry to obtain a continuous pose estimate of the vehicle at all times. This is similar to the manner in which location recognition on a pre-built map is combined with visual odometry in traditional VSLAM techniques.

Our experiments use a real-time system operating in a car to validate the capabilities of our method. We compare our results to a state of the art VSLAM method [11] to demonstrate the localization accuracy as well as robustness to lighting and appearance change of our method. Our results are favorable even though road marks appear only occasionally and road markings by themselves do not form a complete localization solution but only a first step towards our use of higher-level semantic features. A discussion on the limitations of the current system and future directions for improvements rounds off our contributions in this work.

II. Related Work

Autonomous driving, as well as many driver safety applications, require precise localization to within a couple of meters at most. Current consumer-level GPS devices cannot guarantee this level of accuracy, especially in dense urban areas where a lot of driving takes place [28]. GPS receivers that take into account vehicle dynamics and use 3D models of the environment to predict signal attenuation can improve accuracy to some extent but do not solve the problem completely [18].

The most reliable existing approach to vehicle localization is through the use of the Velodyne lidar sensor, which was widely used in DARPA challenges [16], [27], and is also used by the Google driverless car. One approach has been to build a 2D map of the environment by projecting the point cloud onto the road surface [10]. Map building is performed offline by fusing high accuracy GPS measurements with IMU, and lidar. GPS imprecision is mitigated by map matching [9]. The resulting map is globally precise to within a few centimeters. However, drawbacks of the method include the high cost of the sensor and the data intensive nature of the maps.

We are interested in obtaining precise localization using computer vision. While the above approach using lidar could, in principle, be transferred to vision-based systems, this poses a number of practical problems. It is more convenient and more robust to build sparse feature-based maps using vision in contrast to the dense grid maps that can be built using lidar. However, the features used are sensitive to lighting and appearance so that reliable localization is much more complicated than using a lidar. Approaches to reliable localization over long periods of time include the use of high resolution panoramic images [29], learning appearance change in feature descriptors [15][21], and creating a new map whenever localization fails [3]. The last approach, where each new map is called an "experience", is based on knowing what road the car is driving on from other sensors such as GPS. The main drawback is the requirement by every vehicle to collect data, build maps online, and share these maps. This may be impractical for large scale use but may be a good approach for reliable localization on a few specific routes, such as for a specific commute [4]. Our approach is somewhat different as it is directed towards having specially equipped "mapping" vehicles that use an expensive high accuracy GPS, while all other vehicles simply localize using pre-built maps, cameras and low cost GPS sensors.

A few existing methods use higher level features for precise localization on roads. Senlet and Elgammal use satellite images to segment roads [24] and sidewalks [25] to precisely localize vehicles and robots respectively. However, this method does not work when trees or tall buildings obstruct the satellite view. Another similar approach that matches lane and road markings between in-vehicle camera and recorded aerial imagery to perform localization is [20]. [14] uses realistic models of intersections to generate simulated data that is compared to the actual camera image. The intersection modeling requires manual labeling and is tedious. Our focus in this paper, in contrast to above methods, is to select features that are highly distinctive and can be mapped while driving a vehicle on the road. Our entire system, and the resulting map representation, is automatic and efficient. While road marks alone cannot provide a complete localization solution due to their sparse occurrence on roads, our use of them is a step towards this general goal.

III. Road Marking Detection

We begin by providing an overview of our road marking detection and recognition algorithm which forms the basis of our localization system. The road marking detector (RMD) detects and recognizes the type of road mark by learning feature-based templates of the markings using training images. Templates are learned from training images which contain manually annotated bounding boxes for the road markings. Each road marking type (Stop, Bike lane, turn arrows etc.) may have many templates corresponding to various views, road and lighting conditions. The training images are first rectified to compensate for lens and perspective distortions, the latter in particular being done with an inverse perspective transform [2]. Then, we use MSER [13] to find regions of interest that could potentially contain road markings. Since MSER detects regions of high contrast, this type of use is apt in this case and corresponds to a robust version of image binarization. Subsequently, we detect FAST corners [22] within the regions of interest. The corner features, their HOG descriptors [5], and the label of the road mark are stored as the template information that is used for detection during runtime.

At runtime, inverse perspective rectification, MSER detection, FAST corner detection, and HOG descriptor computation are performed on each test image. The signs in the testing images are then detected and identified based on the corner features. First, we find putative matching pairs of corners based on their HOG descriptors. Subsequently, we refine the result through a structural matching algorithm that matches the 2D geometry of the corners within the road marking. The geometry matching takes into account the possibility

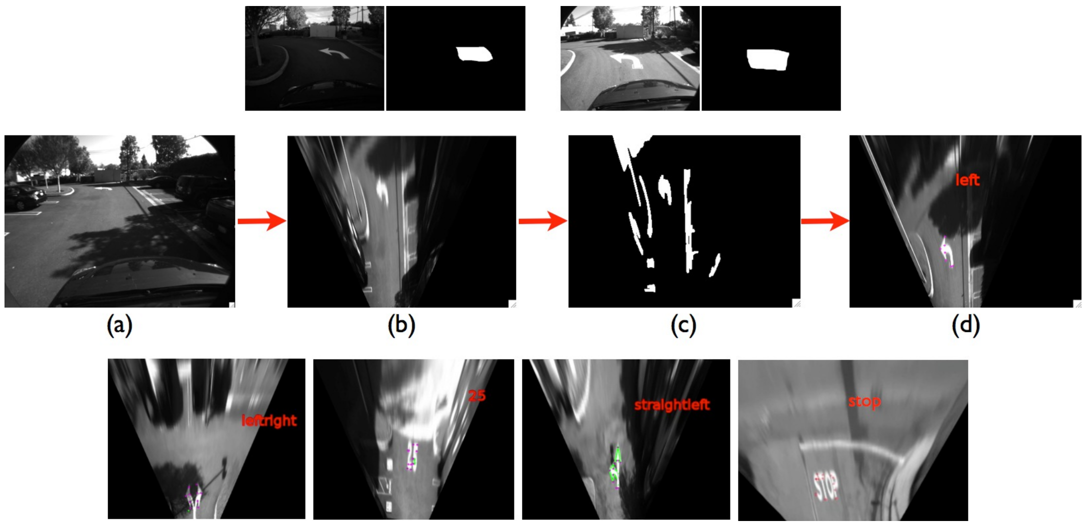

Fig. 1. Road marking recognition: (top row) Two examples of template images for a specific road mark type along with the manually annotated ground truth masks. (middle row) Data flow of the algorithm - the input image (a) is converted into the inverse perspective mapped (IPM) image (b) on which we detect MSER regions as a means of robust binarization. FAST corners detected in the MSER regions are matched to corresponding features detected in the similarly transformed template images to detect the road mark and simultaneously classify it based on the label associated with the matched template (shown in (d)). (bottom row) Examples of road marking detections. The detection algorithm can handle different lighting conditions, shadows, and some amount of occlusion.

of multiple road markings in the image as well as the possibility of some features in a template not being detected due to changes in lighting and perspective. The reason for employing feature point-based matching rather than a full shape matching is to ensure robustness to occlusion and partial shadows on the road marks, in which case some features or part of the shape may not be detectable. An algorithm flow of the road marking recognition is given in Figure 1. The system is highly reliable at detection, having an accuracy of more than 91% on a dozen types of road marks. More details of the algorithm and an evaluation of its performance can be found in [30].

Pose Estimation from Road marks

Pose estimation is based on the corner features detected within the landmark. The global coordinates of these corners have to be known to perform pose estimation, and these are mapped as explained in Section IV. However, this implies that the surveyed points have to repeatably detected in the reliable manner within the road mark. We do this using the following process. During the mapping phase, the corners detected on the road mark that have been matched to the corresponding template image are selected and surveyed. We detect the contour of the road mark on the inverse perspective mapped (IPM) image using an active snake algorithm [8] and record the relative pixel locations of these corners within the contour. During localization, the same process is repeated and only the corners at the same relative locations on the shape contour as the surveyed corners are picked.

The snake algorithm is typically used to find the contour of a given shape represented by a set of points, in our case the corner features. The algorithm iteratively finds locations for the points that minimize an energy function and lock on to a suitable contour. In our case, the contour is the road marking and the contrast in illumination between the markings and the road is sufficient for the algorithm to work reliably. More details of the snake algorithm are provided in [8].

The instantaneous position of the camera on observing a road mark is obtained by assuming that the global coordinates of the FAST corners detected on the road mark are known. Since the detection is done on the inverse perspectively mapped (IPM) image, let the positions of the corners in the IPM image be denoted as $\left[U, V\right]$ and their physical locations in global coordinates as $\left[X, Y\right]$ where U, V, X, Y are $n \times 1$ vectors, and n is the number of corners detected in the road mark. We fit an affine transformation $A \equiv \begin{bmatrix} Q & t \\ 0 & 1 \end{bmatrix}$ between these two 2D points sets.

$$Q, t = \underset{Q,t}{\arg\min} \| \left[X, Y\right]^T - (Q \left[U, V\right]^T + t) \|_2^2 \qquad (1)$$

Now, in addition, if we assume the camera is extrinsically calibrated, i.e. we know the height and mounting angle of the camera on the car, then the pixel location (u_{car}, v_{car}) of the camera in the IPM image is also known (though it will lie outside the image). Applying the affine transformation A to (u_{car}, v_{car}) yields the 2D global coordinates of the camera (x_{car}, y_{car}).

$$\left[x_{car}, y_{car}\right]^T = Q \left[u_{car}, v_{car}\right]^T + t \qquad (2)$$

The 2D rotation contained in the affine transformation corresponds to the global yaw angle. Hence, the 2D camera pose can be estimated very efficiently using only a monocular camera. A drawback of the method is the assumption of a flat ground or, at least, known pitch of the road, and also the requirement of the extrinsic calibration of the camera. However, as we demonstrate, the method provides poses with high accuracy.

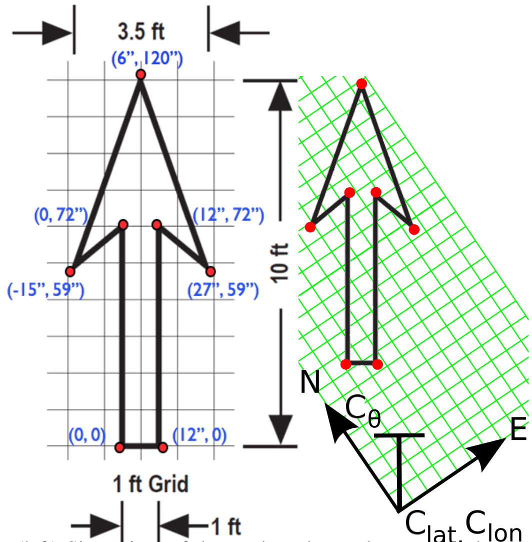

Fig. 2. (left) Since sizes of the road marks are known exactly, corners on the road mark can be assigned coordinates in a local coordinate system with absolute scale in meters. (right) Mapping to global coordinates. The point (u_{car}, v_{car}) coincides with (C_{lat}, C_{lon}).

IV. Mapping and Localization using Road Marks

In our case, mapping implies surveying the coordinates of the corner points in the road marks. While this could be done in an arbitrary coordinate system, use of GPS coordinates provides an accurate way of measuring large areas. During mapping, we use the combination of a highly accurate GPS receiver and a high-end IMU, the filtered 6DOF output of which is accurate to within 20cm in position and 0.5^o in each rotation angle.

Since road marks are standardized, their sizes are fixed and known (for instance, in [19]). With a monocular camera setup, we use these known sizes to calibrate the scale s of absolute distances in the IPM image. Let the global position and yaw for the camera computed from the GPS/IMU output be given as $C_{lat}, C_{lon}, C_\theta$. Then equation 3 transforms pixel coordinates (u, v) to meters such that the y-axis is aligned with due North and the x-axis with due East (see figure 2). Equation 4 then gives an excellent local approximation for the GPS coordinates of each corner on the road mark.

$$(x_m, y_m) = sR(C_\theta)((u, v) - (u_{car}, v_{car})) \tag{3}$$

$$\begin{aligned} y_{lat} = & \quad C_{lat} + y_m/111111.111 \\ x_{lon} = & \quad C_{lon} + x_m/(cos(C_{lat}) * 111111.111) \end{aligned} \tag{4}$$

Apart from the coordinates of the corners themselves, we store the road mark label (forward arrow, left turn, yield etc) with the GPS location of the centroid. These are used for look-up into the map when localizing. If two or more road marks of the same type are present in an image, for instance in Figure 3, we also store with the label a numeric identifier increasing from left to right and top to bottom to disambiguate the road marks.

During localization in our current system, we run visual odometry using a calibrated stereo camera. We use the

Fig. 3. When multiple road marks are present in an image, they are disambiguated by adding a numeric identifier to the label.

Fig. 4. Localization under change in lighting - We test and compare our system using two datasets with significant lighting change between them (top) dataset under good daylight conditions collected around 11am (bottom) dataset collected close to sundown.

algorithm given in Lim et. al. [11] for this purpose. The visual odometry algorithm uses Harris corners that are tracked using a KLT tracker that also includes an epipolar constraint to discard spurious feature tracks. 3D locations of the features are initialized by using triangulation. Relative motion is computed between keyframes using the 3-point algorithm [6]. Keyframes are created based on a threshold on translational and rotational motion. We use a windowed bundle adjustment to smooth the noise due to visual odometry, for which purpose we use the sparse bundle adjustment algorithm [12].

Pose estimation from road marks is used to correct drift in visual odometry by incorporating the absolute pose computed from the road mark into the windowed bundle adjustment. We create a keyframe whenever a road mark is detected and include the pose obtained from the road mark as a measurement in the bundle adjustment. This allows drift correction from instantaneous pose estimates in global coordinates.

V. Experiments

We present experiments using our system running live in a test car. The car is equipped with an Oxford Inertial+ IMU and Navcom SF-3050 GPS receiver, which provides accurate 6DOF pose information that we use for mapping and ground truth. We use a stereo pair of PointGrey Grasshopper cameras with a baseline of 90cms mounted on top of the car and looking to the front for computing visual odometry. The images from the left camera are used as input to the

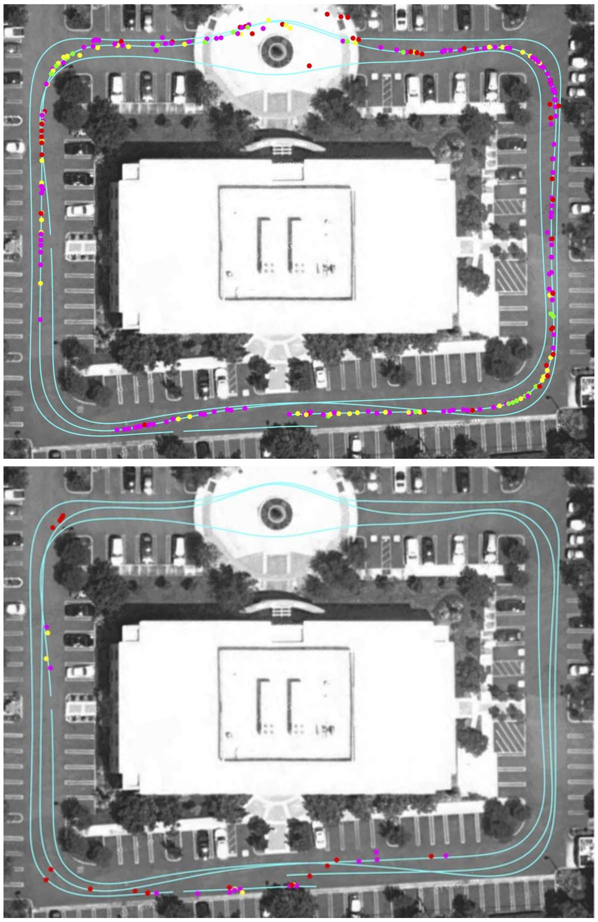

Fig. 5. Typical loss of localization accuracy under large appearance change (here lighting) using the OEM method [11] - Shown are overhead views with trajectory of vehicle during collection of reference images indicated by cyan curves. Dot locations show position estimates when recognition occurs. Color indicates 2D error; green: <0.5m, yellow: <1m, magenta:<3m, red: >= 3m. Top image shows results where lighting conditions are similar to those when reference images were collected. Bottom image shows results with poor lighting during recognition (just before sundown). Both frequency of detection and accuracy of pose estimation suffer noticeably in the second example.

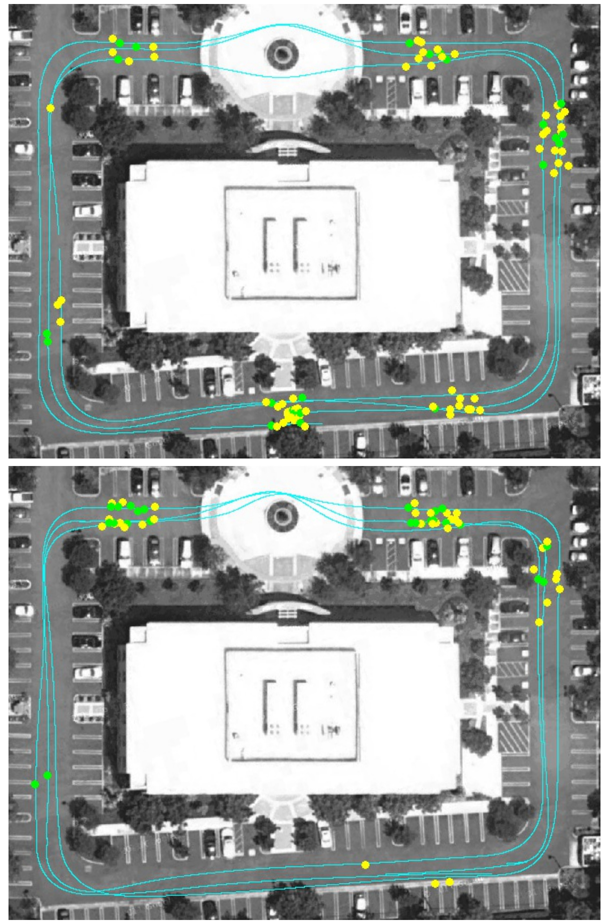

Fig. 6. Robustness of pose estimation from road markings - Accuracy of localization is preserved under lighting change. Dot locations show pose estimates from road marks. Colors used correspond to those in Figure 5. The top image shows results where lighting condition is similar to those of the template images. Bottom images shows results with poor lighting (just before sundown). Though frequency of detections reduces in certain areas, accuracy is maintained.

road marking detection and pose estimation algorithm. The entire system uses only grayscale images. The GPS/IMU system provides data at 50Hz while the camera resolution used is 640x480. The system runs at upwards of 10Hz with the feature detection and tracking implemented on a GPU.

We compare our localization result to the "Online Environment Mapping" (OEM) system of Lim et. al. [11]. We build maps that are registered to ground truth GPS using the OEM system. This is done by associating each keyframe with the corresponding GPS measurement during the bundle adjustment process that produces the optimized global map. We localize against a pre-built map of the environment using a vocabulary tree [17]for image retrieval followed by geometric matching of features to verify the image match. SURF descriptors computed at corner features are used for the vocabulary tree queries. The relative pose between the reference keyframe from the map and the current keyframe is obtained using the 3-point algorithm within a RANSAC estimator for robustness. More details of the algorithm can be found in [11].

Our experiment consists of data collected in the driveway around a building, the length of the loop being around 300 meters. We collected data at two different times of the day, once at 11am and once at 5pm. Sample images from both datasets are shown in Figure 4 where significant lighting change is evident. The OEM map was built using the 11am dataset. Six road marks of different types were set up at the various locations along the loop to enable testing of the road mark pose estimation. The templates for the road mark were collected at the same time as the 11am dataset. Three templates were collected from different distances and viewing angles for each road mark. Ground truth GPS data was collected at both times. The parameters of the visual odometry and the road marking detector were manually optimized for good performance on the 11am data while the location recognition and pose estimation parameters for localizing using the OEM algorithm were optimized to provide good pose estimates when localizing on the 11am map using the same dataset. We then tested localization by taking each dataset as the test sequence in turn.

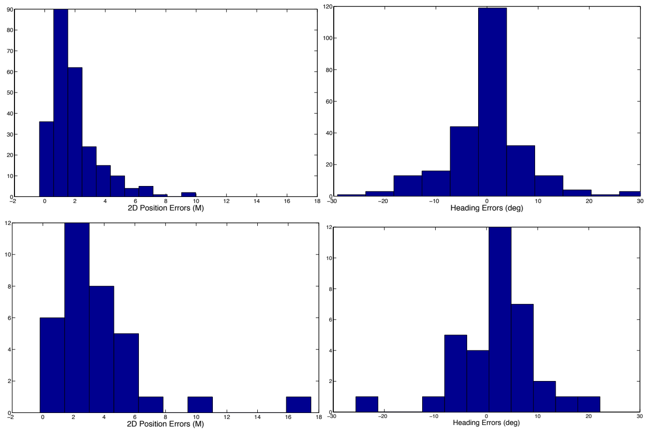

Fig. 7. Histograms of 2D position error and heading error for localizing using an OEM map under the same test and map-building conditions (top) and different lighting conditions (bottom). General spread of error remains the same although number of pose estimations against the map reduces drastically. Y-axis of the histograms represents pose estimation count.

The results are presented in three parts. First, we compare the accuracies of pose estimates by localizing on an OEM map and using road marks. The pose estimates obtained when this procedure is successful are visualized in Figure 5 on the ground truth GPS trajectory. As can be seen clearly from the figures, pose estimates are consistent and relatively accurate when localizing under exactly the same conditions, which is the best case when using an OEM map. However, the estimates degrade drastically when trying to localize using the 5pm dataset (note that the same map built using the 11am data was used for matching). This is to be expected as most of the features detected at 11am cannot be reliably detected at 5pm, and further, even those that can be detected can often not be matched using the OEM method under such pronounced lighting changes. This results in fewer and less accurate pose estimates. In contrast, the corresponding estimates from road marks, shown in Figure 6, are generally more accurate than OEM map-based localization even when lighting conditions are the same. When conditions are different, as shown at the bottom of Figure 6, the estimates maintain their accuracy although a slight decrease in the number of detections is noticeable. Based on this, we can conclude that lighting change affects our system significantly less than traditional VSLAM techniques, resulting in greater robustness and practicality.

The second part of our results consists of a quantitative analysis of the performance of the two algorithms. The position and heading error histograms for localizing on an OEM map in both lighting conditions are shown in Figure 7. Note that while the number of location recognitions and pose estimation reduces significantly, the overall distribution of error remains almost the same. The reason for this is that there are very few "false positives" in location recognition, i.e. a location is almost never confused with another, at least in our datasets where there is very little aliasing. In a few cases, a place is "recognized", a few meters before or after the reference location resulting in increased number of outliers at the edges of the histogram. The heading, however, remains consistent.

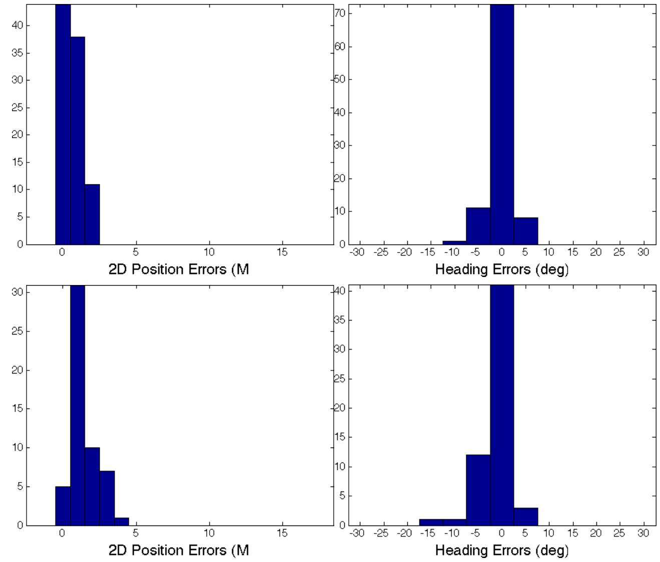

Fig. 8. Histograms of 2D position error and heading error for localizing using road marks under the same test conditions as the template images (top) and different test conditions (bottom). Accuracy is much higher than for map-based localization in both cases and accuracy is maintained across the two cases. The number of detections and pose estimation also does not drop as drastically as with map-based localization. Y-axis of the histograms represents pose estimation count.

The error histogram for road mark-based pose estimation (Figure 8) reveals the higher accuracy of the method by more than a factor of two. Further, while performance remains almost constant across the two different test conditions, the number of pose estimates also does not decrease significantly. This is in contrast to the OEM map-based localization results.

VI. Discussion

We presented a system for light-weight localization using road markings for obtaining instantaneous pose estimates. Reliable corners estimated within the road marks, which are detected in turn using a specialized road marking detector, are used to obtain precise localization estimates. The main motivation for using road marks is to avoid the problems caused by trying to localize on a sparse feature map under lighting and appearance change. We verified that this problem is largely ameliorated by our method in our experiments where we compared a VSLAM system running visual odometry and location recognition against the map with our light-weight localization system that uses visual odometry and absolute pose estimates from road marks. The localization accuracy of our system, when compared to ground truth GPS, is at least as good as localizing using the OEM method even in conditions most favorable for the latter method. When conditions are unfavorable for this method, i.e. when the appearance between the map conditions and the test conditions is very different, localization accuracy falls off rapidly. In contrast, our approach is largely unaffected even by significant lighting change, as is expected.

Our system runs at approximately 10 Hz in a car. The main computational constraint in the road marking detection algorithm is the feature-based template matching, which scales linearly with the number of templates. However, this step could be easily parallelized as each template can be matched independently of the others. We envision our method as a first step towards the more general use of high-level features for vehicle localization. Such maps would be light-weight and would support more robust localization under changing conditions. The ultimate goal is to build a map once and localize against at any time in the future. This is not possible with point feature-based maps currently in wide use.

The major drawback of our method is, of course, that it is only applicable to localization on roads with clearly painted markings. The complexity of the map has been pushed into the feature detection (road mark detection, in our case) instead of being in the representation itself. In addition, road marks are frequently occluded in traffic so that any method depending solely on them will not be practical. We intend to address this by including other types of high-level features.

Other shortcomings that we intend to address in future work include the assumption of flat ground needed to create the inverse perspective mapped (IPM) image. One way to address this would be through the use of an IMU to provide instantaneous pitch and roll angles of the vehicle. Currently, we do not also address inconsistencies in GPS between mapping runs due to changing reception conditions. The resulting map has to be generated by reconciling the two sets of measurements (as, for instance, is done in [10]). Another requirement of the current method is the need to have the exact shapes of the road marks. These shapes have to be obtained manually which is a bit tedious although not unimaginable since the total number of road marks is not very large. The need for knowing the shape and size of the road marks can be overcome by using stereo cameras to triangulate the points and obtain their 3D location or by using the planar assumption with a monocular camera setup that has scale calibrated by other means. This is also part of future work.

REFERENCES

[1] Motilal Agrawal and Kurt Konolige. Frameslam: From bundle adjustment to real-time visual mapping. *IEEE Transactions on Robotics*, 24(5), October 2008.

[2] M. Bertozzi, A. Broggi, and A. Fascioli. Stereo inverse perspective mapping: theory and applications. *Image and Vision Computing*, 8(16):585–590, 1998.

[3] Winston Churchill and Paul Newman. Continually improving large scale long term visual navigation of a vehicle in dynamic urban environments. In *Proc. IEEE Intelligent Transportation Systems Conference (ITSC)*, Anchorage, USA, September 2012.

[4] Winston Churchill and Paul Newman. Practice makes perfect? managing and leveraging visual experiences for lifelong navigation. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Minnesota, USA, May 2012.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 886–893, 2005.

[6] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[7] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J.J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering. In *IEEE Intl. Conf. on Robotics and Automation, ICRA*, pages 3281–3288, Shanghai, China, May 2011.

[8] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *INTERNATIONAL JOURNAL OF COMPUTER VISION*, 1(4):321–331, 1988.

[9] Kurt Konolige. Large-scale map-making. In *AAAI*, pages 457–463, 2004.

[10] J. Levinson, M. Montemerlo, and S. Thrun. Map-based precision vehicle localization in urban environments. In *Proceedings of Robotics: Science and Systems*, 2007.

[11] J. Lim, J.-M. Frahm, and M. Pollefeys. Online environment mapping. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[12] M.I. A. Lourakis and A.A. Argyros. SBA: A Software Package for Generic Sparse Bundle Adjustment. *ACM Trans. Math. Software*, 36(1):1–30, 2009.

[13] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference*, pages 414–431, 2002.

[14] N. Mattern and G. Wanielik. Camera-based vehicle localization at intersections using detailed digital maps. In *2010 IEEE/ION Position Location and Navigation Symposium (PLANS)*, pages 1100–1107, 2010.

[15] Andrej Mikulík, Michal Perdoch, Ondřej Chum, and Jiří Matas. Learning a fine vocabulary. In *European Conference on Computer Vision*, 2010.

[16] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt, and S. Thrun. Junior: The stanford entry in the urban challenge. *Journal of Field Robotics*, 25(9):569–597, 2008.

[17] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[18] Marcus Obst, Sven Bauer, Pierre Reisdorf, and Gerd Wanielik. Multipath detection with 3d digital maps for robust multi-constellation gnss/ins vehicle localization in urban areas.

[19] California Dept. of Transportation. California manual on uniform traffic control devices, 2012 edition, 2012.

[20] O. Pink. Visual map matching and localization using a global feature map. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPRW '08*, pages 1–7, 2008.

[21] A. Ranganathan, S. Matsumoto, and D. Ilstrup. Towards illumination invariance for visual localization. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2013.

[22] E. Rosten, R. Porter, and T. Drummond. Faster and better: A machine learning approach to corner detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32:105–119, 2010.

[23] S. Se, D. Lowe, and J. Little. Global localization using distinctive visual features. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 226–231, 2002.

[24] T. Senlet and A. M. Elgammal. A framework for global vehicle localization using stereo images and satellite and road maps. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 2034–2041, 2011.

[25] Turgay Senlet and Ahmed Elgammal. Satellite image based precise robot localization on sidewalks. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2012.

[26] Gabe Sibley, Christopher Mei, Ian Reid, and Paul Newman. Adaptive relative bundle adjustment. In *Robotics Science and Systems Conference*, pages 1–8, 2009.

[27] C. Urmson, J. Anhalt, D. Bartz, M. Clark, T. Galatali, A. Gutierrez, S. Harbaugh, J. Johnston, H. Kato, P. L. Koon, W. Messner, N. Miller, A. Mosher, K. Peterson, C. Ragusa, D. Ray, B. K. Smith, J. M. Snider, S. Spiker, J. C. Struble, J. Ziglar, and W. (Red) L. Whittaker. A robust approach to high-speed navigation for unrehearsed desert terrain. *Journal of Field Robotics*, 23(8):467–508, August 2006.

[28] US Dept. of Defense and GPS Navstar. Global positioning system standard positioning service performance standard, 2008.

[29] Christoffer Valgren and Achim J. Lilienthal. Sift, surf and seasons: Appearance-based long-term localization in outdoor environments. *Robotics and Autonomous Systems*, 58(2):157–165, February 28 2010.

[30] Tao Wu and A. Ranganathan. A practical system for road marking detection and recognition. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 25 –30, june 2012.