2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Online Approximate Model Representation of Unknown Objects

Kiho Kwak, Jun-Sik Kim, Daniel F. Huber, and Takeo Kanade

***Abstract*— Object representation is useful for many computer vision tasks, such as object detection, recognition, and tracking. Computer vision tasks must handle situations where unknown objects appear and must detect and track some object which is not in the trained database. In such cases, the system must learn or, otherwise derive, descriptions of new objects. In this paper, we investigate creating a representation of previously unknown objects that newly appear in the scene. The representation creates a viewpoint-invariant and scale-normalized model approximately describing an unknown object with multi-modal sensors. Those properties of the representation facilitate 3D tracking of the object using 2D-to-2D image matching. The representation has both benefits of an *implicit* model (referred to as a view-based model) and an *explicit* model (referred to as a shape-based model). Experimental results demonstrate the viability of the proposed representation and outperform the existing approaches for 3D-pose estimation.**

I. Introduction

Object recognition and tracking are well-studied problems in computer vision [1] [2]. Such algorithms frequently learn a representation of an object or object category from a database of labeled training examples [3] [4] [5], though some methods can recognize objects with few examples or even just one [6] [7]. In situations where novel objects need to be detected and tracked, learning methods that require training examples are not applicable. For example, if an autonomous vehicle has a vision system trained to detect and track people and other vehicles, how will the system perform if it encounters a cow crossing the road? While it may be possible to recognize the cow as an obstacle without any training examples, it would be beneficial to be able to reliably track the object from a distance and to predict its future motion.

This paper describes a method for creating a model of an object online as it is observed and then demonstrates how this model can improve detection and tracking of the object when it is observed again in the future. The choice of representation of the model can have a significant impact on the capabilities and performance of a vision algorithm. A representation should describe the shape and appearance of 3D objects [8]. Historically, object representations follow one of two extremes for encoding geometry. Explicit representations model the 3D shape of an object, for example, by creating a point cloud or surface mesh model of the object [9] [10]. Implicit representations model shape indirectly without explicitly encoding 3D shape, for example, by storing a set of representative images of an object taken from different viewpoints [11] [12]. Explicit representations have the advantage that they can compactly represent an object and can handle novel viewpoints, but creating an accurate 3D model is in itself a challenging task. Implicit representations can be easier to create, but they require significant storage and can have difficulty with novel viewpoints.

In this paper, we advocate an alternative representation that takes an intermediate position between explicit and implicit representations, which we call an *approximate model*. Rather than explicitly attempting to model all the details of an objects shape, we distill the geometry down to a small number of planar, vertically-oriented patches, each of which is coupled with an appearance model for that patch. Figure 1 shows an overview of tracking an object using an approximate model representation.

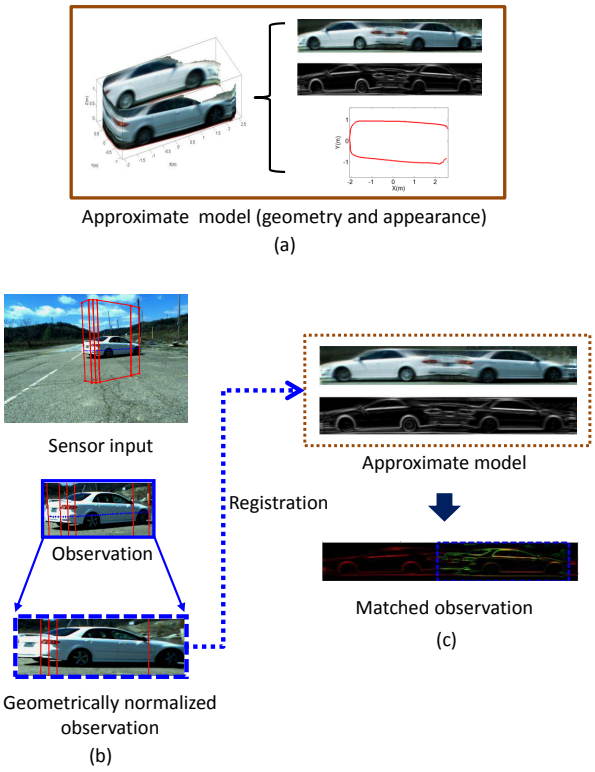

Fig. 1. Tracking using the proposed approximate model. (a) The approximate model is created online from a sequence of observations of the target object. (b) A new view of the object is observed and automatically segmented from the image. (c) The new observation is matched to the approximate model to determine the matching location and orientation of the target object.

Kiho Kwak is with Agency for Defense Development, Republic of Korea kkwak.add@gmail.com

Jun-Sik Kim is with Korea Institute of Science and Technology, Republic of Korea junsik.kim@kist.re.kr

Daniel F. Huber and Takeo Kanade are with Carnegie Mellon University, Pittsburgh, PA 15213, USA {dhuber@ri.cmu.edu, tk@cs.cmu.edu}

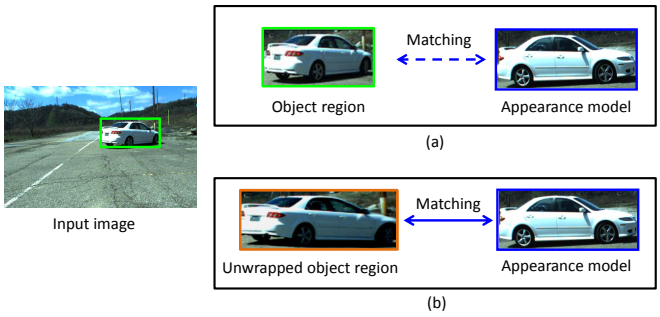

Fig. 2. Appearance matching. The green box shows the Region Of Interest (ROI) of the tracked object in a frame. The blue box shows the appearance model of the object which is obtained from the first frame. (a) The dashed line describes that the image matching between the model and the observation is difficult. (b) The line shows that the unwrapped image makes the image matching easy.

We demonstrate approximate modeling with an algorithm that builds an approximate model online from a sequence of observations of a moving object taken from a moving platform. Our modeling approach uses a combination of 2D imagery and and range imaging (from a single-line LIDAR). The images provide dense appearance information but cannot easily provide accurate shape information, while the LIDAR provides sparse shape information but limited appearance data. The combination of the two sensors gains the benefits of both.

Creating an approximate model from a sequence of observations is challenging for several reasons. First, the model must be created from scratch. In order to relate the current observation to previous ones, it is necessary to determine the correspondence between the new observation and the in-progress model. Our approach operates incrementally, incorporating new observations as they arrive, continuously improving the approximate model. Segmentation algorithms are used to automatically disambiguate the target object from background regions.

Second, we observe that finding correspondences between views can be challenging due to viewpoint-dependent appearance changes, scale changes, and illumination variations from shadows (Figure 2). Our approximate model representation addresses these challenges using an *unwrapped image*, which is a scale-normalized, fronto-parallel image of the object (Figure 1(a)). The image is created using coarse geometry provided by 3D imaging to warp the observed image to approximate how it would appear from a frontal viewpoint. In this way, unwrapped images from different viewpoints are transformed to become more similar (Figure 1(b)). To obtain robustness to illumination changes, we match images using a combination of color similarity and image gradients (edge strength)(Figure 1(c)). The unwrapped image improves image matching under appearance variations due to illumination, viewpoint, and partial occlusion and improves detection and tracking performance.

We illustrate the benefits of approximate models through experiments that compare detection and tracking performance to alternative methods. Our experiments show that we can track novel objects more accurately and at longer distances using the approximate model.

The contributions of this paper are threefold. 1) We introduce the concept of approximate modeling. 2) We demonstrate approximate modeling with an online algorithm for creating approximate models of moving objects observed from a moving platform. 3) We demonstrate the benefit of approximate modeling in real-world experiments using a system integrated onto an autonomous vehicle testbed.

II. Approximate Representation in a Single Frame

In a single frame image, we can approximately represent the object with the corresponding geometry. Our representation approximates the object shape with a set of piecewise planar patches, and each patch contains a scale-normalized and fronto-parallel image of the object as a descriptor. Two preliminary works, extrinsic calibration of a lidar and a camera and segmentation and detection of moving object regions, are achieved by using the algorithms in [13] and [14], respectively.

Our approximate representation approach in a single frame consists of two steps: shape representation with a piecewise linear model and appearance representation using the approximate shape. These steps are described in the following subsections.

A. Shape Representation with Piecewise Linear Model

Given raw lidar measurements of an object (Figure 3 (a)), we first filter out some unreliable measurements using the Rahmer-Douglas-Peucker algorithm (referred to as Iterative End Point Fit algorithm) [15] [16]. The algorithm, given a curve composed of a set of points, finds a similar curve with fewer points. Using the algorithm, we roughly fit the raw lidar measurements to a piecewise linear curve L composed of line segments.

Once we obtain L from the raw lidar measurements, we compute the number of line segments in L. Each line segment implies a 3D vertical planar patch composing the object surface, as we approximate the shape with a set of vertical planar patches. Estimating the object geometry ($\hat{L} = \{\hat{L}_i\}, i = 1, \cdots, n$) composed of n line segments is achieved by using the Douglas-Peucker algorithm (Figure 3 (b)).

B. Appearance Representation using Approximate Shape

Each fitted line $\hat{L}_i$ of a lidar geometry directly corresponds to a plane of an object's surface. Once we know the corresponding image region that is the object ROI I with the lidar geometry, we divide the object ROI into n sub-image regions I_i.

In order to unwrap the object image I (a perspective distorted image) using $\hat{L}$, we should estimate a homography matrix $\mathbf{H}$ which is a transformation from I to the unwrapped object image plane $\hat{I}$ (Figure 4). Building $\hat{I}$ composed of n sub-unwrapped image planes, we compute each homography $\mathbf{H}_i$ from I_i to $\hat{I}_i$. Let I_i have four vertices $\mathbf{p}_i^j = (u_i^j, v_i^j, 1)^T$, $j =$

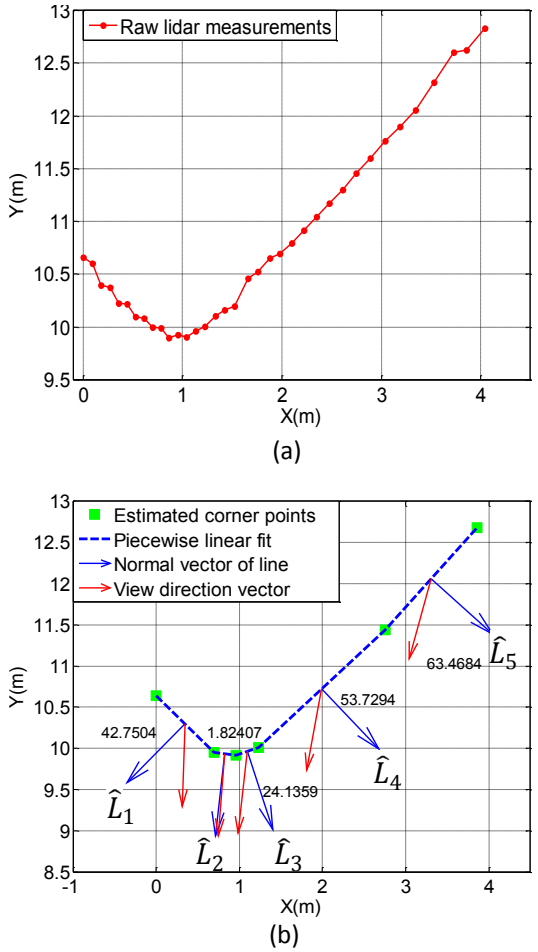

Fig. 3. Fitting lidar measurements. (a) The raw lidar measurement is obtained from a moving car. The red dots show the return points from the surface of the car. (b) Each blue dashed line between the estimated corner points (the end points of each line segment) shows the approximated plane segment that composes the surface of the car. The number of line segments means that the object surface consists of five planes. The numbers on each line segments show the angle difference between the normal direction vector (blue arrows) of each line and the viewing direction vector (red arrows) from the sensor.

$1,\cdots,4$ in homogeneous coordinates and the corresponding vertices $\mathbf{q}_i^j = (x_i^j, y_i^j, 1)^T$ on $\hat{I}_j$ exist. $\mathbf{H}_i$ is estimated as [17]:

$$\mathbf{q}_i^j = \mathbf{H}_i \mathbf{p}_i^j. \tag{1}$$

To decide the corresponding vertices $\mathbf{q}_i^j$, it is required to know the size of the unwrapped object image plane $\hat{I}$. It is decided by the spatial resolution per pixel α $(mm/pixel)$ and the height of the object region. In this paper, we set the object height to a predefined height h because we do not know the real object height yet. Let l_i be the real length of the line segment. The number of pixels in a row and a column of $\hat{I}_i$ are computed by $\frac{h}{\alpha}$ and $\frac{l_i}{\alpha}$, respectively. The spatial resolution α is dependent on the working distance D that can be decided by the sensor's accuracy. If D increases, α also increases:

$$\alpha = \frac{2D\tan(H_{FOV}/2)}{H_{res}}, \tag{2}$$

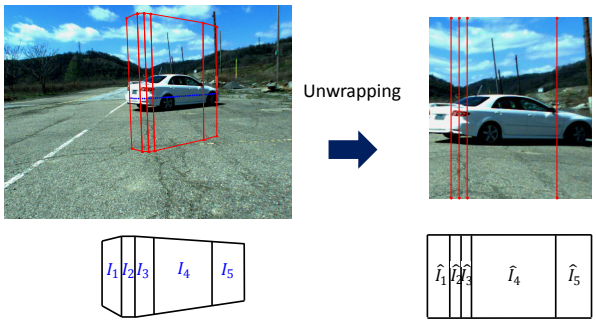

Fig. 4. Unwrapped image planes using the corresponding lidar measurements. The sub-image planes (I_1, I_2, I_3, I_4, I_5) in the object ROI are estimated with the fitted lines ($\hat{L}_1, \hat{L}_2, \hat{L}_3, \hat{L}_4, \hat{L}_5$) obtained by piecewise linear fitting, respectively. Height of the imaged planes is fixed to a predefined height (6 m). The unwrapped image $\hat{I}$ is obtained by stitching the unwrapped images ($\hat{I}_1, \hat{I}_2, \hat{I}_3, \hat{I}_4, \hat{I}_5$).

where H_{FOV} is the horizontal camera field of view (radians) and H_{res} is the horizontal camera resolution (pixels).

III. Approximate Modeling from a Sequence of Observations

Once we create the unwrapped object image while keeping a track of a previously unseen object, the approximate model of the object is created by incrementally registering the unwrapped object images. To register the images, we track the object over time, and then align the tracked object images. The tracked object images are extracted by the boundary detection algorithm in [14] that is formulated as a classification problem that determines whether an object boundary exists between two consecutive range measurements.

We align a current unwrapped image of an object onto the reference image updated with previous images. Since the unwrapped images are invariant to scale and viewpoint changes, the unwrapped images are registered by 2D-to-2D appearance matching (Figure 5). In order to minimize the difference between the reference image and the current image, we update the reference image by integration of measurements using a new mosaicing approach that applies higher weights to measurements from the closet and the most frontal viewpoint.

Our approximate modeling from a sequence of observation of a moving object is achieved with three steps: registration of unwrapped images, temporal integration of unwrapped images, and foreground region estimation. These steps are described in the following subsections.

A. Registration of Unwrapped Images

Registering two unwrapped object images is ideally a simple 1D matching problem in the horizontal direction because the unwrapped images are scale-normalized at fixed vertical location of the lidar scan plane. In practice, because the ground is not perfectly flat, we allow a little vertical movement in the registration.

To estimate quality of alignment of the two unwrapped images, we ensure the weighted sum of correlation values of

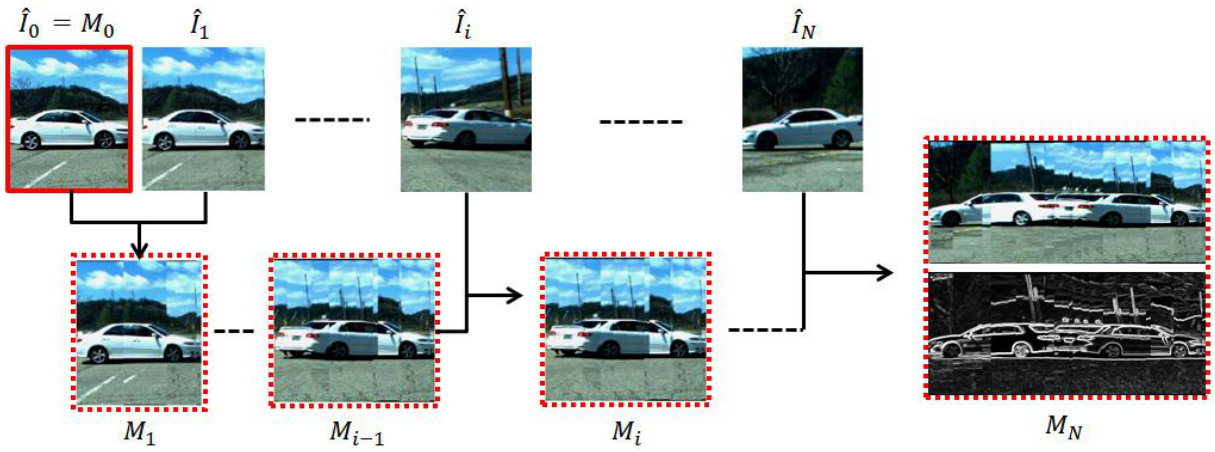

Fig. 5. Updating the reference image using integrated mosacing. Red boxes show the reference images that we obtained at each frame. Each reference image consists of color appearance and edge-gradient.

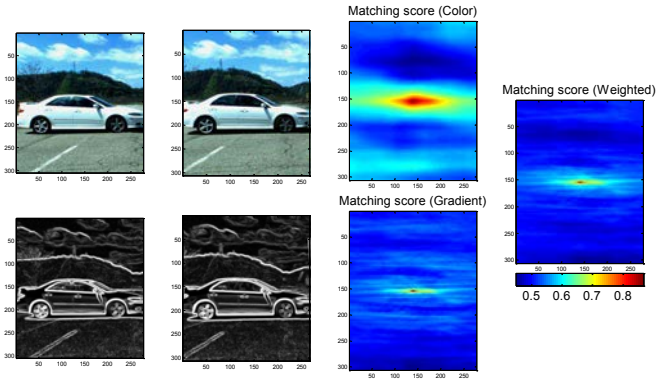

Fig. 6. Correlation of two unwrapped object images. Both the test images are obtained with two-frame difference (15 frames/second). In the correlation score, blue is low (< 0.5) and red is high values (> 0.8). The three correlation scores indicate that both of the images are almost overlapped ; the point with the highest score is at the center of the score map. The score map of the color appearances has a large variance along the horizontal direction. Although the edge-gradients seem to be well correlated with a smaller variance, the background edges affect to the correlation score. The weighted sum of both the two score maps shows that the problems of the appearance and edge-gradient matching are solved. The weight of the correlation score of the color appearance is 0.3.

color appearance and edge-gradient, as shown in Figure 6. In outdoor environments, object's appearance is influenced by various factors, such as illumination, brightness, and shadows. Specifically, the color appearance is more seriously affected than the edge-gradient by those factors [18]. Thus, we apply different weights to the alignment of the color appearance and the edge-gradient.

Given a current image $\hat{I}_t(\mathbf{x})$ sampled at a discrete pixel location $\mathbf{x}_i = (x_i, y_i)$, registration is to find where it is located in the previous image $\hat{I}_{t-1}(\mathbf{x})$. The correlation value is computed as following [19]:

$$S(\mathbf{u}) = \frac{\sum_i \{\hat{I}_{t-1}(\mathbf{x}_i) - \bar{\hat{I}}_{t-1}\}\{\hat{I}_t(\mathbf{x}_i + \mathbf{u}) - \bar{\hat{I}}_t\}}{\sqrt{\sum_i \{\hat{I}_{t-1}(\mathbf{x}_i) - \bar{\hat{I}}_{t-1}\}^2 \{\hat{I}_t(\mathbf{x}_i + \mathbf{u}) - \bar{\hat{I}}_t\}^2}}, \quad (3)$$

where $\mathbf{u} = (u, v)$ is the displacement, and

$$\begin{aligned} \bar{\hat{I}}_{t-1} &= \frac{1}{N}\sum_i \hat{I}_{t-1}(\mathbf{x}_i) \\ \bar{\hat{I}}_t &= \frac{1}{N}\sum_i \hat{I}_t(\mathbf{x}_i + \mathbf{u}). \end{aligned} \quad (4)$$

are the mean intensity of the corresponding patches and N is the number of pixels in the patch. Once we compute the correlation scores of the color appearance $S_c(\mathbf{u})$ and $S_g(\mathbf{u})$ of $\hat{I}_t(\mathbf{x})$ and $\hat{I}_{t-1}(\mathbf{x})$, the best alignment is made at the position of which the weighted sum of both correlation scores is maximized:

$$\mathbf{x}^* = \arg\max_{x,y} \{wS_c(\mathbf{u}) + (1-w)S_g(\mathbf{u})\}, \ \forall (u,v) \in (x,y). \quad (5)$$

B. Building Reference Image: Temporal Integration of Unwrapped Images

Registration aligns the corresponding unwrapped images from multiple views of a tracked object, and building a single reference image can be made by integrating them properly. In order to integrate the aligned unwrapped images, we incrementally build a mosaic by assigning the column images of the selected views onto the mosaic image plane. Since the part of the appearance model are usually observed integrated mosaic is typically visible in multiple frames, column images of each of these assigned images are potential candidates in the integrated mosaic. We have two challenges in creating the integrated mosaic incrementally: (1) how to consider an artifact problem occurred by different illumination conditions, (2) how to minimize the difference between the integrated mosaic and a new unwrapped image when we register both images.

Compositing the integrated mosaic M is performed by optimally choosing source images for the mosaic using a graph cut optimization [20]. Let $\hat{I}_1, \cdots, \hat{I}_m$ be the set of aligned unwrapped images to the mosaic image plane. The graph cut estimates a label image F in which the label x at

column q denoted by F_q indicates that an image $\hat{I}_x$ should be used as the source for qth column in the mosaic.

The energy function that we minimize is denoted by $E(F)$ that is the sum of a data penalty term expressing the quality of the set of aligned unwrapped images used in the mosaic and a pairwise interaction penalty term explaining the discontinuities of all pairs of neighboring columns in the mosaic. Let $\mathscr{P}$ be the set of all columns in M and $\mathscr{N}$ be the set of all adjacent column pairs in M:

$$E(F) = \sum_{q \in \mathscr{P}} D_q(F_q) + \sum_{(q,r) \in \mathscr{N}} V_{q,r}(F_q, F_r), \tag{6}$$

where the data penalty term denoted by $D_q(F_q)$ is the cost of assigning label F_q to column q and is defined by the plane weights as:

$$D_q(F_q) = w_{F_q}(q), \tag{7}$$

The weight w of the unwrapped image $\hat{I}$ is obtained by estimating the angle ϕ between the sensor's viewing direction and the normal direction of the object surface using the corresponding lidar data $\hat{L}$, so that the plane image captured at the most frontal viewpoint is preferable. Let $\hat{I}$ be composed of n plane segments and the size of $\hat{I}$ be $r \times c$. The weight w_k, $k = 1, \cdots, c$ of each column is estimated by using the angle ϕ [21] [22]:

$$w_k = \sin^2 \phi_k. \tag{8}$$

The pairwise interaction penalty term $V_{q,r}(F_q, F_r)$ is the cost of assigning label F_q and F_r to neighboring columns q and r in the label image. This is to avoid selecting too different images for nearby columns. Note that, if $F_q = F_r$ then $V_{q,r}(F_q, F_r) = 0$.

$$V_{q,r}(F_q, F_r) = \|F_q - F_r\|, \tag{9}$$

C. Foreground Region Estimation

The integrated mosaic benefits from registering the unwrapped images incrementally. However, the mosaic is not directly able to be used as a model because we do not estimate the foreground in the mosaic and there are many artifacts produced by stitching the unwrapped object images obtained from different lighting condition and vignette effects.

To remove these artifacts and estimate the foreground region in the object ROI, we obtain the mean appearance by computing the average values of colors and gradient magnitudes of the registered unwrapped images (see Figure 7 (a)). The mean appearance benefits from estimating the foreground region. As the object moves, the background in the object ROI will typically change more dramatically than the foreground object appearance. Therefore, the foreground region is estimated by detecting the point where the background variability begins to occur. This also means that the consistency of edge gradient in the background region is not guaranteed but that in the foreground is consistent

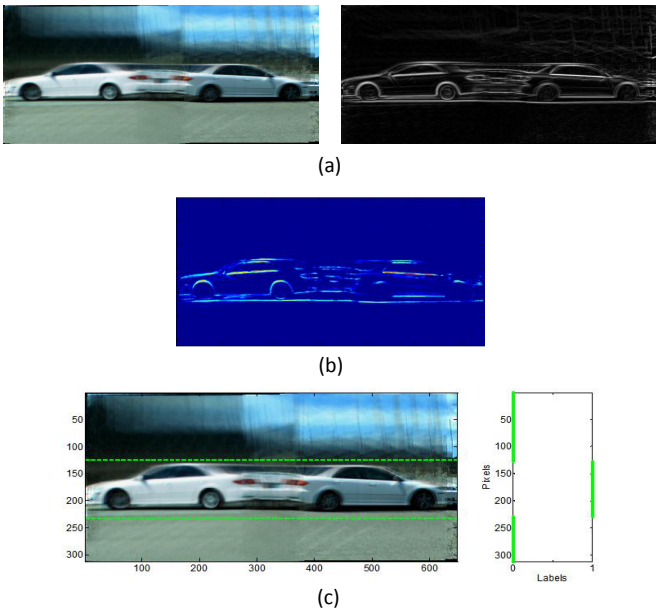

Fig. 7. Horizontal boundary estimation using an MRF optimization. (a) shows the mean color appearance obtained with the registered unwrapped object images. The object is tracked for 65 frames. (b) describes the confidence of edge gradient. Blue denotes values no greater than zero and other colors describe positive confidence values. Closer to red corresponds to higher confidence, closer to blue to lower confidence. (c) shows the estimated horizontal boundaries (green) in the mean color appearance. The foreground rows in the image are labeled as 1 and the others are the background as 0. In the right figure, x-axis is the label values and y-axis is the row index of vertical pixels. (Best viewed in color)

(see Figure 7 (b)). We pose the foreground region detection as a horizontal boundary detection problem using an MRF optimization. The horizontal boundary detection is a kind of binary labeling problem which determines whether a row is the foreground or not (see Figure 7 (c)).

1) Gradient Confidence of Registered Images: Suppose we have two unwrapped object images ($\hat{I}_i$, $\hat{I}_j$) which are already registered. The consistency of edge gradient between $\hat{I}_i$ and $\hat{I}_j$ is achieved by measuring the confidence of gradient vector field [18]. Let G_i and G_j be the gradient vector fields of $\hat{I}_i$ and $\hat{I}_j$ respectively. Given the matching position $\mathbf{x}^* = (x^*, y^*)$, the gradient confidence C of the intersection region between the both images is estimated by:

$$\begin{aligned} C(\hat{I}_i \cap \hat{I}_j) \quad = \quad & \frac{1}{2}(\|G_i(x^*, y^*)\| + \|G_j(x^*, y^*)\|) \\ & - \|G_i(x^*, y^*) - G_j(x^*, y^*)\|, \end{aligned} \tag{10}$$

the confidences except the intersection region are computed by:

$$C(\hat{I}_i - (\hat{I}_i \cap \hat{I}_j)) = \frac{1}{2}\|G_i(x^*, y^*)\|, \tag{11}$$

$$C(\hat{I}_j - (\hat{I}_i \cap \hat{I}_j)) = \frac{1}{2}\|G_j(x^*, y^*)\|. \tag{12}$$

2) Horizontal Boundary Estimation: We use graph cut optimization to detect the horizontal boundaries in the mean appearance. The graph cut optimization returns the label (e.g., $\{Foreground = 1, Background = 0\}$) of each row minimizing an energy function with two energy terms. The first energy term corresponds to the consistency of edge gradient in the mean appearance and the second energy term relates with the color similarity between row pixels in the mean appearance. Let $\mathscr{V}$ be the set of all rows in the gradient confidence map C and $\mathscr{N}$ be the set of all adjacent row pairs in the mean color appearance $\bar{I}_c$. This binary labeling problem assigns a label F_p to pth row. The labeling variables F is obtained by minimizing the energy function as Equation 6.

The data penalty term $D_p(F_p)$ is defined as the negative log likelihood of the foreground and background confidence distributions of the row of C. Let the gradient confidence map C be $m \times n$ and $\hat{C}_+$ and $\hat{C}_-$ be the sum of the number of positive and negative confidences in C. The foreground and background confidence distribution are obtained as:

$$\begin{aligned} p(\hat{C}|Foreground) &= \frac{1}{n}\hat{C}_+ \\ p(\hat{C}|Background) &= \frac{1}{n}\hat{C}_-. \end{aligned} \tag{13}$$

Then, we use the distributions to set $D_p(F_p)$ as negative log likelihoods:

$$D_p(F_p) = \begin{cases} -\ln p(\hat{C}|F_p = 1) & \text{if } F_p \text{ is foreground} \\ -\ln p(\hat{C}|F_p = 0) & \text{if } F_p \text{ is background} \end{cases} \tag{14}$$

The pairwise term $V_{p,s}(F_p, F_s)$ is penalizes the appearance differences between adjacent rows p and s in $\bar{I}_c$. This term is defined as:

$$V_{p,s}(F_p, F_s) = |F_p - F_s| \cdot \exp(-D_{p,s}). \tag{15}$$

where $D_{p,s}$ is the similarity of the two image pairs computed by using Histogram Intersection [19]. Given a pair of histograms, $\mathbf{H}_p$ and $\mathbf{H}_s$, of pth and sth images respectively, each containing k bins, the histogram intersection of the normalized histogram is defined as follows:

$$D_{p,s} = \sum_{i=1}^{k} \min\left(\mathbf{H}_p(i), \mathbf{H}_s(i)\right), \tag{16}$$

where, we map the colors in the image channel into a discrete color space containing k bins.

IV. Experiments

A SICK LMS-221 lidar and a PointGrey Flea2 camera were used to acquire the data sets for our experiments. The lidar has a 180 degree horizontal field of view, with a line scanning frequency of 75 Hz and a 0.5° angular resolution. The camera has a 60° horizontal field of view, with a frame rate of 15 Hz and a resolution of 1024 by 768 pixels. These sensors were mounted on front of a vehicle. The lidar was placed at a height of 70 cm, and the camera was installed 30 cm above the lidar. The data from each sensor was time-stamped to enable synchronization.

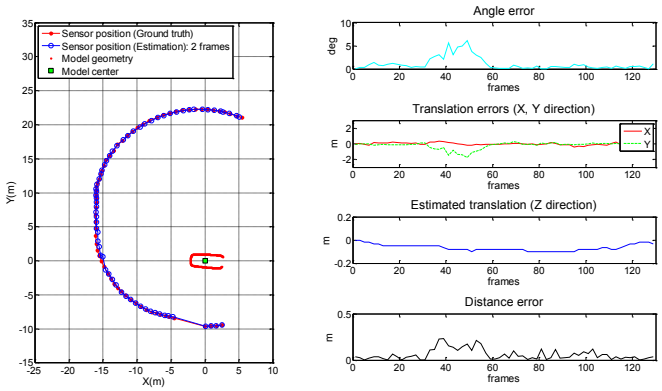

Fig. 8. Evaluation of 3D sensor pose estimate. The frame rate is 7.5 frames/second; that is, we use 65 frames of data with 2 frame intervals out of 130 data frames. The initial few frames are artificially generated for evaluating the robustness of our algorithm.

Using the framework described above, we performed a series of experiments to evaluate the effectiveness of our approximate representation approach. In the first experiment, we evaluated the accuracy of the 3D sensor-pose estimation using our approximate representation algorithm. We then compared our representation algorithm to two state-of-the-art methods: iterative closet point (ICP) [23] and the color ICP approach [24]. Finally, we evaluated the detection and tracking performance of our approximate model.

A. Evaluation of 3D Pose Estimation Accuracy

We evaluated the accuracy of the 3D sensor pose estimate. The groundtruth pose data for comparison has been obtained by manual alignment. The performance was evaluated in three aspects: angle error, translation error on the XY-ground plane, and distance error between the ground-truth position and the estimated position. Additionally, we estimated the translation in the Z direction. The testing was achieved with the two sets running on different frame rates, i.e., two- and five-frame intervals. The object views vary more dramatically as the frame interval is increased.

Figure 8 shows the results of 3D sensor pose estimation. Our approach accurately estimates the sensor position independent of the frame rate. As shown in Figure 8, the average of the angle error is 1.2° at the two frame rate. The translation errors in the Y direction on the XY-ground plane increase increases at the specific frames. Angular error causes a large error in the Y translation direction because the angle error increases at the specific frames where the sensor position moves on the Y axis. Since we use lidar geometry for the modeling, the maximum distance error is less than $0.3m$. The largest angle errors occur when the object moves on the slanted ground because our approximate representation does not estimate roll and pitch motion.

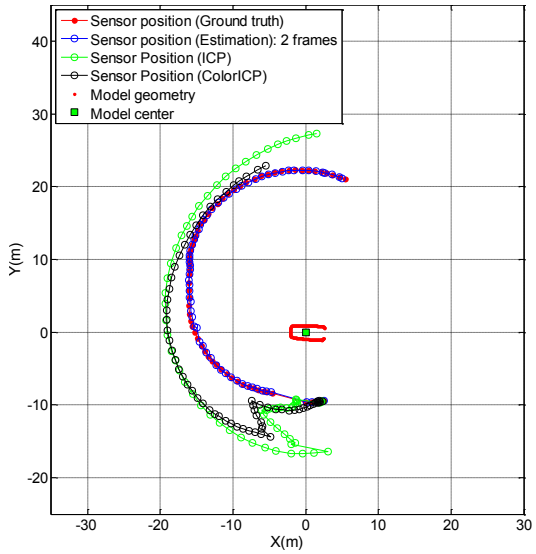

Fig. 9. Comparison with ICP and color ICP. The color ICP algorithm performs better than the ICP algorithm in the case of abrupt view change, but color ICP also does not overcomes the estimation error of the sensor position in the case of featureless data sets.

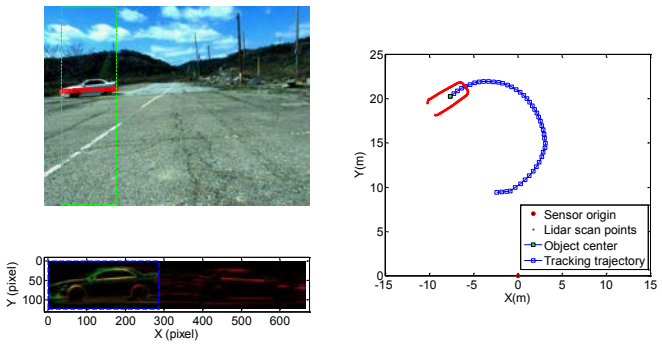

Fig. 10. Object tracking with 3D pose estimation. In the scene image, the green box represents the object image region that is detected using the vertical boundary detection approach. The geometric model of the tracked object is shown as a red in the right image. The image in the lower left corner shows the matching region on the model of the tracked object. The red shows the edge gradients of the object model, the green and yellow describe the edge gradient matched with high probability.

B. Comparison against Existing Methods

In our second experiment, we compared the performance of our algorithm with two existing algorithms; the iterative closet points (ICP) approach and color ICP. Both algorithms are state-of-the-art approaches to register or align data, though there are many variants of these. The color ICP considers not only point data, but color as well. For the color ICP approach, the texture information was used with small image patches centered on the projection of lidar measurements.

The results, shown in Figure 9, show that our algorithm outperforms both algorithms for the initial frame section of 1 to 40 frames. These frames consist of challenging cases such as abrupt view changes and featureless geometry data. The misalignment of the ICP algorithm was caused by some featureless geometry data (e.g., lines without corners). The ICP algorithm requires a good initial transformation in order to converge to the globally optimal solution, otherwise only a local optimum is achieved. The color ICP is robust with respect to featureless data, but the corresponding image patches are not discriminative because most cars have textureless surfaces and similar colors at the same height.

C. Evaluation of Detection and Tracking Performance

We test the performance of our detector and the tracking accuracy with 3D pose estimation of unknown object using our approximate model. For the experiment, we first create the approximate model of an object of interest using half of the frames in a full frame data set and then detect and track the object in the full frames. Figure 10 shows the tracking result of an object in 3D space. For these experiments, we detect an object at 3 frame intervals in 130 frame data and estimate the 3D pose of the object. The tracking was performed by detecting the object and estimating the 3D pose. The tracking trajectory was obtained by tracking the estimated center of mass of the object. We predicted an accurate trajectory in all frames.

V. Summary and Future Work

We have presented an algorithm approximately representing unknown objects in outdoor environments using lidar measurements and corresponding imagery. The success of the proposed method lies in three central stages which consist of approximate representation in a single frame, online approximate modeling from a sequence of observation of a moving object, and real-world experiments using the approximate model.

The approximate model consists of the unwrapped mosaics (i.e, color and edge-gradient) and the approximate geometry of an object. These model components are created by an incremental mosaicing with unwrapped images which are invariant to viewpoint and scale changes. With the approximate model, we detect and track an object in a scene, and then estimate its 3D pose by 2D-to-2D matching between the unwrapped observation and its approximate model. The proposed algorithm is tested with the datasets we obtained in an outdoor environment. The approximate representation outperforms both ICP and color ICP alignment algorithms for estimating the sensor position in 3D. The detecting and tracking results shows that the proposed representation improves on the performance achieving the both benefits of the 2D and 3D model-based approaches by detecting the object and estimating its 3D pose through 2D image matching.

In future research, we hope to address a number of issues. First, we would like to evaluate the detection and tracking performance of the proposed approach on more complicated paths of unknown objects. Second, we plan to extract models for other classes such as pedestrians or bicycles using the proposed algorithm.

REFERENCES

[1] J. Mundy, "Object Recognition in the Geometric Era: A Retrospective," in *Toward Category-Level Object Recognition*, ser. Lecture Notes in Computer Science, J. Ponce, M. Hebert, C. Schmid, and A. Zisserman, Eds. Springer Berlin Heidelberg, 2006, vol. 4170, pp. 3–28.

[2] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Computer Survey*, vol. 38, no. 4, pp. 13+, Dec. 2006.

[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2005, pp. 886–893.

[4] B. Leibe, K. Schindler, N. Cornelis, and L. Van Gool, "Coupled object detection and tracking from static cameras and moving vehicles," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, pp. 1683–1698, October 2008.

[5] A. Ess, K. Schindler, B. Leibe, and L. Van Gool, "Object detection and tracking for autonomous navigation in dynamic environments," *The International Journal of Robotics Research*, vol. 29, pp. 1707–1725, December 2010.

[6] N. Dowson and R. Bowden, "Simultaneous modeling and tracking (smat) of feature sets," in *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 2, june 2005, pp. 99 – 105 vol. 2.

[7] Z. Yin and R. Collins, "On-the-fly object modeling while tracking," in *Computer Vision and Pattern Recognition. IEEE Computer Society Conference on*, june 2007, pp. 1 –8.

[8] Y. Sato, M. D. Wheeler, and K. Ikeuchi, "Object shape and reflectance modeling from observation," in *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH '97, 1997, pp. 379–387.

[9] D. Koller, K. Danilidis, and H.-H. Nagel, "Model-based object tracking in monocular image sequences of road traffic scenes," *International Journal of Computer Vision*, vol. 10, pp. 257–281, June 1993.

[10] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce, "3d object modeling and recognition using affine-invariant patches and multi-view spatial constraints," in *Computer Vision and Pattern Recognition. IEEE Computer Society Conference on*, vol. 2, june 2003, pp. II – 272–7 vol.2.

[11] H. Schneiderman and T. Kanade, "A statistical method for 3d object detection applied to faces and cars," in *Computer Vision and Pattern Recognition, Proceedings. IEEE Conference on*, vol. 1, 2000, pp. 746–751 vol.1.

[12] A. Torralba, K. Murphy, and W. Freeman, "Sharing features: efficient boosting procedures for multiclass object detection," in *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 2, 2004, pp. II–762–II–769 Vol.2.

[13] K. Kwak, D. Huber, H. Badino, and T. Kanade, "Extrinsic calibration of a single line scanning lidar and a camera," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[14] K. Kwak, D. Huber, J. Chae, and T. Kanade, "Boundary detection based on supervised learning," in *Robotics and Automation, IEEE International Conference on*, 2010.

[15] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica The International Journal for Geographic Information and Geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.

[16] V. Nguyen, S. Gächter, A. Martinelli, N. Tomatis, and R. Siegwart, "A comparison of line extraction algorithms using 2d range data for indoor mobile robotics," *Autonomous Robots*, vol. 23, no. 2, pp. 97–111, Aug. 2007.

[17] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[18] D. Scharstein, "Matching images by comparing their gradient fields," in *The International Conference on Pattern Recognition*, 1994, pp. 572–575.

[19] R. Szeliski, *Computer Vision : Algorithms and Applications*, R. Szeliski, Ed. Springer-Verlag New York Inc, 2010.

[20] Y. Y. Boykov and M. P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images," in *International Conference on Computer Vision*, vol. 1. IEEE Comput. Soc, 2001, pp. 105 – 112.

[21] V. S. Lempitsky and D. V. Ivanov, "Seamless mosaicing of image-based texture maps," in *Computer Vision and Pattern Recognition. IEEE Computer Society Conference on*. IEEE Computer Society, 2007.

[22] S. N. Sinha, D. Steedly, R. Szeliski, M. Agrawala, and M. Pollefeys, "Interactive 3d architectural modeling from unordered photo collections," *ACM Transactions on Graphics*, vol. 27, no. 5, pp. 159:1–159:10, Dec. 2008.

[23] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *International Journal of Computer Vision*, vol. 13, no. 2, pp. 119 – 152, Oct. 1994.

[24] A. E. Johnson and S. B. Kang, "Registration and integration of textured 3-d data," in *Proceedings of the International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, ser. NRC '97. IEEE Computer Society, 1997, pp. 234–.