2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

A Robust Aim Point Tracking Algorithm for 3-D Laser Radar Imagery

Romeil Sandhu, Shawn Lankton, Samuel Dambreville, Scot Shaw
Dan Murphy, Allen Tannenbaum

Abstract— **In this work, we present a controlled active vision tracking scheme that makes use of 3-D range data and 2-D reflectance data collected with a 3-D Laser Radar (3DLADAR) system. Specifically, our algorithm employs the Active Contour Framework along with a detection algorithm involving feedback to effectively maintain visual tracking of an object in adverse scenarios. As opposed to typical 2D systems, in which tracking is limited by a lack of contrast, 3DLADAR provides the capability to improve aim point tracking by encoding depth information so that the scene can be resolved for all spatial dimensions. We demonstrate the proposed algorithm both qualitatively and quantitatively on several challenging tracking scenarios.**

I. Introduction

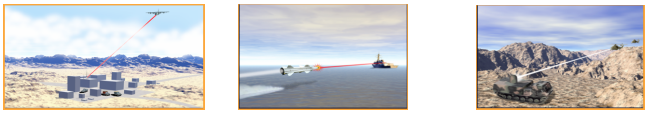
A well studied problem in controlled active vision is the fundamental task of visually tracking a deformable object in an adversarial environment [4], [15]. In this work, we present a new tracking scheme using imagery taken with the 3-D Laser Radar (3DLADAR) system developed at MIT-Lincoln Labs [1]. Specifically, our algorithm employs the Active Contour Framework [5], [9], [6] as well as a detection algorithm based on appearance models to efficiently track a "target" under challenging scenarios. As opposed to typical 2D systems [16], [2], in which tracking is limited by a lack of contrast, 3DLADAR provides the capability to improve aim point tracking by encoding range and angle-angle information. Typical environments in which 3DLADAR can be used are shown in Figure 1. Consequently, we can tackle the problem with the following two pronged approach:

- First, use image segmentation in a coarse manner to capture 'target" features.
- Second, use a point set registration scheme to align features to a template model.

However, to appreciate the contributions presented in this paper, we briefly revisit some results that have been made pertaining to the field of visual tracking which specifically invokes principles of feedback and control theory.

The first class of visual tracking schemes propose the use of a finite dimensional representation of continuous curves. Specifically, the B-Spline representation is often used for the "snake" model [15]. From this, Isard and Blake propose the Condensation algorithm [4]. Similarly, with the use of B-Splines and assuming a unimodal distribution of the state vector, the unscented Kalman Filter is introduced for rigid object tracking [7], [11]. A major drawback associated with these tracking schemes is the fact that they track only a finite dimensional group of a target parameters, and do not explicitly handle the local deformation of an object.

Recent work has addressed this by using an implicit representation of the contour via level set methods to handle the local deformation of the object [10], [6]. In particular, Yezzi and Soatto decouple motion into two distinct parts: a "global" rigid motion and a "deformation", which is given by any departure from rigidity [17]. This enables the algorithm to appropriately track through clutter and occlusions. In a similar fashion, Rathi et al. [12] use particle filtering in conjunction with active contours to track deformable objects. While their approach is able to track objects in more challenging sequences, it becomes increasingly computationally burdensome. To this end, [8] propose an unscented Kalman Filtering approach with the additional step of learning shape information prior to tracking. We note that while our method shares similarities with the above tracking schemes, namely the use of active contours, one fundamental difference is that we do not require the notion of multiple hypothesis to account for clutter, occlusions, and erratic behavior. Instead, we treat these artifacts as events that are to be captured via appearance models, which are learned online. Moreover, the multiple hypothesis nature of both [12], [8] are generally not ideal for the 3DLADAR imagery at hand, and are likely to fail when facing occlusions. This is because the background in 3D range data is not as distinguishable as in pure 2D imagery, which causes a poor estimate of the object's location. Lastly, our algorithm requires no "off-line" step such as shape learning and is able to cope with various targets.



(a) Air-Ground (b) Horizontal (c) Ground-Air

Fig. 1. Several Environments for which 3DLADAR is employed.

The remainder of this document is organized as follows: In the next section, we introduce the mathematics and concepts associated with active contour segmentation as well as the notion of appearance models. In particular, we derive an entirely new segmentation technique to specifically handle range imagery. In Section III, we describe the integration and development of the active contour tracker (ACT) in a systematic manner involving feedback to tackle the problem of 3DLADAR tracking. We then demonstrate the robustness of the algorithm to turbulence and occlusions on several challenging tracking scenarios in Section IV. Finally, we give concluding remarks in Section V.

978-1-4577-0079-8/11/$26.00 ©2011 AACC

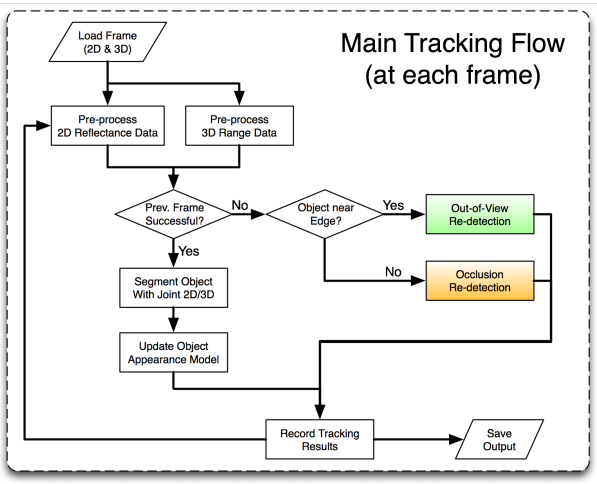


Fig. 2. Flow chart describing overall operation of the tracker.

II. Preliminaries

In this section, we introduce two new concepts for this paper: Segmentation with Thresholding Active Contours (TAC) and Appearance Models. These concepts will be essential for the proposed tracking algorithm.

A. Thresholding Active Contours (TAC)

To define the TAC, consider an image I over the domain $\Omega \in \mathbb{R}^2$ that is partitioned into regions by an evolving contour C, where C is embedded as the zero level set of a signed distance function $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$ such that $C = \{x|\phi(x) = 0\}$ [14], [10]. The shape of the evolving contour is described by the Heaviside function, $H\phi$, which is 1 when $\phi < -\epsilon$, 0 when $\phi > \epsilon$, and has a smooth transition in the interval $[-\epsilon, \epsilon]$. Similarly, the interface at the zero level set can be denoted by $\delta\phi$, the derivative of $H\phi$, which is 1 when $\phi = 0$ and 0 at distance ϵ from the interface.

To incorporate statistical information into the segmentation, let us denote the probability of a point being located inside or outside of the curve as G_{in} or G_{out} respectively. Furthermore, we assume that these probabilities are Gaussian: $G_{\text{in}} = \mathcal{N}(\mu_{\text{in}}, \Sigma_{\text{in}})$ and $G_{\text{out}} = \mathcal{N}(\mu_{\text{out}}, \Sigma_{\text{out}})$, where $\mathcal{N}$ denotes the normal distribution. In the case of 3DLADAR, reflectance and range data represent two linearly independent measures. Thus, μ_{in}, μ_{out}, Σ_{in}, and Σ_{out} are each vector valued. Consequently, G_{in} and G_{out} each map $\mathbb{R}^2 \rightarrow \mathbb{R}$. Unlike most segmentation techniques, this statistical information is not used directly, but rather is used indirectly to create a shape model

$$S(x) = H_{\epsilon_2}\left[\log\left(G_{\text{in}}(I(x))\right) - \log\left(G_{\text{out}}(I(x))\right)\right], \quad (1)$$

which serves as a labeling function with ϵ_2 being the threshold for our "estimated" shape. The image shape model S is the most likely shape of the object given current statistical estimates. Note that to mitigate the addition of another parameter, ϵ_2 is simply a scalar multiple of ϵ. From this, the segmenting curve is driven towards this shape by minimizing the following energy

$$E_{\text{image}}(\phi) = \|H\phi - S\|^2 = \frac{1}{2}\int_{\Omega}(H\phi(x) - S(x))^2 dx. \quad (2)$$

Specifically, this energy is the L_2 distance between the shape of the current segmentation and the current estimate of the "correct" shape. Using the calculus of variations, we are then able to compute the gradient $\nabla_\phi \text{E}_{\text{image}}$ as follows:

$$\begin{aligned}\nabla_\phi \text{E}_{\text{image}} = \delta\phi.(H\phi(x) - S(x)) + \beta_\mu^{\text{out}}.\nabla_\phi\mu_{\text{out}} + \\ \beta_\Sigma^{\text{out}}.\nabla_\phi\Sigma_{out} - \beta_\mu^{\text{in}}.\nabla_\phi\mu_{\text{in}} - \beta_\Sigma^{\text{in}}.\nabla_\phi\Sigma_{\text{in}} \quad (3)\end{aligned}$$

where the expressions of the coefficients β_μ^{in} and β_Σ^{in} are given by

$$\beta_\mu^{\text{in}} = \int_\Omega \gamma(u)\Sigma_{\text{in}}^{-1}(I(u) - \mu_{\text{in}})du$$

$$\begin{aligned}\beta_\Sigma^{\text{in}} = \frac{1}{2}\int_\Omega \gamma(u).(\Sigma_{in}^1(I(u) - \mu_{\text{in}})(I(u) - \mu_{\text{in}})^T\Sigma_{\text{in}}^{-1} - \quad (4)\\ \Sigma_{in}^{-1})du \quad (5)\end{aligned}$$

with $\gamma(x) = (H\phi(x) - S(x)).\delta_{\epsilon_2}\left(\log(\frac{G_{\text{in}}(I(x))}{G_{\text{out}}(I(x))}\right)$. In particular, because we deal with 3D range data as well as 2D reflectance data, $\beta_\mu^{\text{in}} \in \mathbb{R}^2$ and $\beta_\mu^{\text{in}} \in \mathbb{R}^4$. Likewise, both β_μ^{out} and $\beta_\Sigma^{\text{out}}$ can be computed by replacing μ_{in} with μ_{out} and Σ_{in} with Σ_{out} in Equation (4). The expression of the gradients for each of the statistical moments are

$$\nabla_\phi\mu_{\text{in}} = \delta\phi.\left(\frac{I - \mu_{\text{in}}}{A_{\text{in}}}\right) \quad , \quad \nabla_\phi\mu_{\text{out}} = \delta\phi.\left(\frac{I - \mu_{\text{out}}}{A_{\text{out}}}\right)$$

$$\nabla_\phi\Sigma_{\text{in}} = \delta\phi.\left(\frac{(I - \mu_{\text{in}})(I - \mu_{\text{in}})^T - \Sigma_{\text{in}}}{A_{\text{in}}}\right)$$

$$\nabla_\phi\Sigma_{\text{out}} = \delta\phi.\left(\frac{(I - \mu_{\text{out}})(I - \mu_{\text{out}})^T - \Sigma_{\text{out}}}{A_{\text{out}}}\right)$$

With the above results, we can now place the image fidelity term in the overall GAC scheme. That is, to minimize this energy via gradient descent, ϕ is updated at each iteration according to

$$\frac{d\phi}{dt} = -\nabla_\phi E_{\text{image}} + \lambda\delta\phi \cdot \text{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) \quad (6)$$

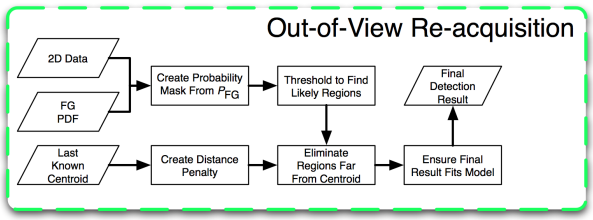


Fig. 3. Flow chart describing the process for re-acquiring objects that have temporarily moved out of view.

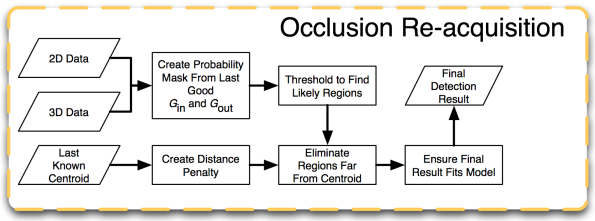


Fig. 5. Flow chart describing the process for re-acquiring objects that have been temporarily occluded.

where the second term in the right-hand-side acts as a regularizing term that penalizes high curvatures. We note that while other active contour energy and segmentation methodologies can be employed, the above scheme leverages the image shape model created from statistical information. This allows for more robust segmentations for 3DLADAR imagery, where under-segmentation is preferred as opposed to over-segmentation. This preference is driven by the fact that each pixel identified as "on-target" in the final segmentation corresponds to a 3D point on the real target. Hence, an under-segmentation ensures that all "on-target" points can be used to understand the shape and pose of the target in world coordinates without being distracted by "off-target" background points.

B. Appearance Models

In a video sequence, we make the assumption of temporal coherency and assume that characteristics of the object will vary slowly over time. With this assumption, we employ appearance models to aid in pre-processing the data, detecting tracking failures, and re-acquiring lost objects.

Three primary aspects of the appearance models exist: Gaussian statistical models, probability density functions (PDFs), and shape information. Gaussian statistical models G_{in} and G_{out}, as described in Section II-A, are stored at each iteration to aid in re-acquisition of the object when it is temporarily occluded. In addition, full intensity histograms of the reflectance data present in the object and background are scaled to produce PDFs P_{in} and P_{out} that are stored at each iteration. This allows the tracker to compare the PDFs of new segmentations with recent segmentations to detect tracking failures and subsequently re-acquire the object. PDFs are always compared using the Bhattacharyya measure,

$$B\big(P_1(x), P_2(x)\big) = \sum_{x=0}^{k} \sqrt{P_1(x) \cdot P_2(x)}dx, \qquad (7)$$

where P_1 and P_2 are any two PDFs [3]. The Bhattacharyya measure is a useful tool because it provides a scalar value in the range $[0, 1]$ corresponding to the similarity of the two PDFs. Finally, shape information such as the area (in pixels) of the object in recent frames are stored to help detect when the object is occluded or moving out of view.

III. Tracking Algorithm

In this section, we discuss the procedure that the tracker follows to continually estimate the 3D location of the object in the 3DLADAR sequence. The tracker combines segmentation and detection in a robust tracking framework. In short, the 3DLADAR data is pre-processed and a joint 2D/3D segmentation is performed at each iteration to find all 3D data points on the target. If necessary, steps are taken to re-acquire lost objects using feedback and exploiting the temporal coherency of the video sequence. Figure 2 summarizes the tracking algorithm from a high level and system perspective.

A. Pre-processing 3D Range Data

As mentioned in the introduction of this note, 3DLADAR offers the capability to resolve all spatial dimensions given the angle-angle and range information. However, unlike typical 2D systems, the notion of an "image" is not immediately available and an image must be formed based on specific assumptions about the acquisition of 3DLADAR data. Moreover, because the 3DLADAR imaging system is not fully developed, we demonstrate results on a simulated 3DLADAR environment, which is pictorially seen in Figure 4.

That is, the 3DLADAR platform delivers several short-pulsed lasers at specific instance of time in hopes of actively illuminating the target. For convenience, we label these series of pulses as a "frame." For each "frame," the corresponding photon arrival time is recorded by the focal plane of an avalanche photo-diode (APD) array. In some cases, arrival times may never be recorded. For instance, if the laser pulse misses the target in clear sky and is never reflected back to the imaging platform. In any event, much of scene will not be recorded for a single frame due to poorly reflected photons. Luckily, the laser pulses are delivered at a pre-specified frequency that is generally much higher than real-time imaging systems.

From this, we are now able to define an "image" as a combination of frames that have been taken over a certain period. Each pixel value of the resulting image is formed by choosing the statistical mode of the values received throughout a series of frames at each pixel location. Additionally, because of the temporal coherency inherent to time-varying imagery, median filtering is performed temporally from image to image to mitigate artifacts caused from the imaging system.

B. Pre-processing 2D Reflectance Data

Processing of the 2D data is also important. Because the object and background may be multi-modal in the

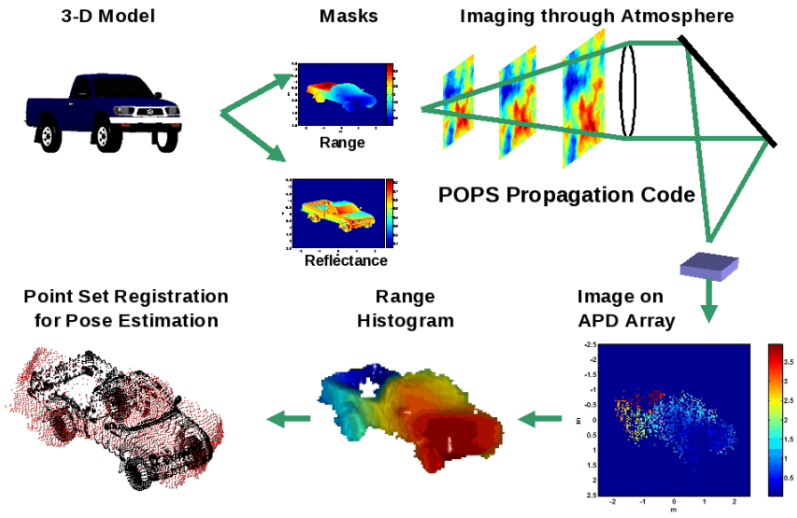


Fig. 4. Simulated Environment for Producing 3DLADAR Imagery with Atmospheric Turbulence.

reflectance data, pre-processing ensures that the object is distinguishable from the background so that segmentation may proceed with ease. First, the PDF of the object (P_{in}) and background (P_{out}) are estimated using the reflectance data and the segmentation result from the previous frame. Next, a foreground histogram, P_{FG} is formed by removing intensity components of the background from intensity components of the foreground,

$$P_{\text{FG}} = \begin{cases} P_{\text{in}} - P_{\text{out}} & (P_{\text{in}} - P_{\text{out}}) > 0 \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

and re-normalizing so that $\|P_{\text{FG}}\| = 1$. Finally, a new image is created corresponding to the likelihood that each pixel is a member of the foreground. Hence, the final result of 2D reflectance pre-processing is

$$\tilde{I}_{\text{ref}}(x) = P_{\text{FG}}(I_{\text{ref}}(x)). \quad (9)$$

C. Re-acquisition (Out-of-View)

If the 3DLADAR device can not accurately follow the movement of the object based on estimates from the tracker, the object may appear to move out of the image domain, and therefore out of the field of view. If this occurs, the procedure shown in Figure 3 is used to detect the object and re-acquire the track once it returns to the FOV.

When searching for the object after it has left view, only the reflectance data is utilized. This is because the object may change its 3D position dramatically during the time it is out of view. Conversely, the reflectance properties of the object should remain constant. Additionally, we assume that the object will reappear near the position (in image coordinates) that it was last seen. One can achieve this by constructing a probability map of a likely "target" via $\mathbb{P}(x) = P_{\text{FG}}(I_{\text{ref}}(x))$.

D. Re-acquisition (Occlusion)

Another failure mode of the tracker occurs when another object in the scene occludes the object of interest, blocking it from view. The procedure shown in Figure 5 is used to re-acquire after the occlusion.

Occlusions of this type are typically shorter and often portions of the object remain visible during most of the occlusion. Hence, we assume that the object will remain at a similar 3D depth and retain its 2D reflectance characteristics. For this case, both types of 3DLADAR data are used. Again, we assume that the object will reappear near the position (in image coordinates) that is was last seen.

To re-acquire the object, the shape model S described in Section II-A is constructed using the statistical models G_{in} and G_{out} from the last successful frame. This shape model is then thresholded to create a binary mask selecting candidate regions that may represent the object. Then, candidate regions that are far from the last known location of the object are excluded and the remaining region is assumed to be the object's current location. If the detected object has a size similar to the object when it was lost, a successful detection has occurred, and tracking will continue normally at the next iteration. Otherwise, this detection procedure will be repeated until the object is successfully re-acquired.

E. Target Appearance Model Update

After segmentation or re-acquisition has occurred successfully, the object's appearance model is updated. This process consists of adding current information such as P_{in}, size of the object (in pixels) and the (x, y) location in image coordinates of the object's centroid to a history of recent values. These values are used to determine appropriate values and thresholds during pre-processing and detection for subsequent frames.

F. Registration and Pose Estimation

Now that segmentation has been performed, a point cloud estimate can be extracted using the specifications of the imaging 3DLADAR system (e.g., slant path, viewing angle). From this, the problem then becomes one of pose estimation or point set registration. In particular, we have experimented

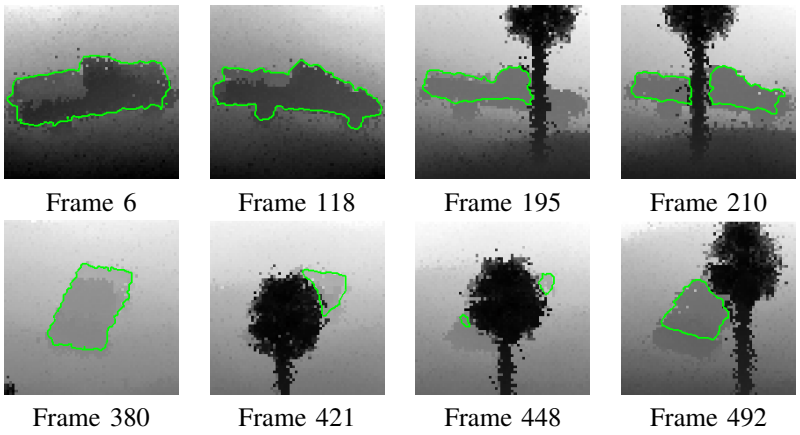


Fig. 6. Visual tracking results of a truck passing through a noisy environment with several varying tree occlusions.

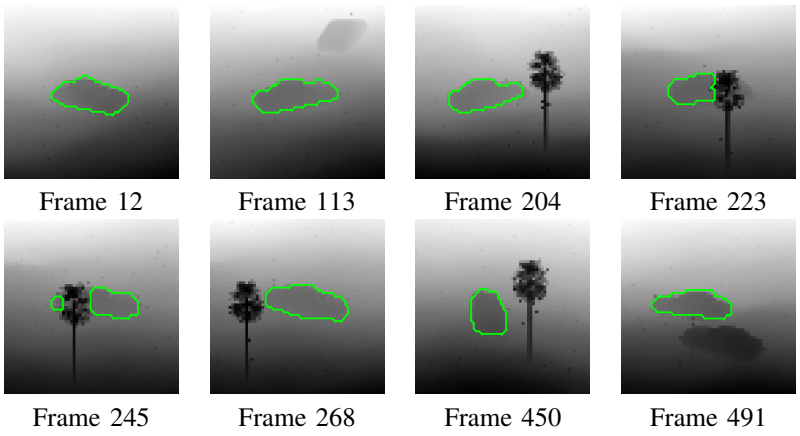


Fig. 7. Visual tracking results of a car passing through a clutter environment that includes complete occlusions as well as non-cooperative targets.

with both the ICP methodology as well a variant of ICP using particle filtering [13]. Although the corresponding pose results were obtained at MIT-Lincoln Labs and are not present in this work, we should note the pose of the target remained sufficiently accurate throughout all tracking scenarios.

IV. Experiments

In this section, we demonstrate the algorithm's robustness to turbulence, occlusions, clutter, and erratic behavior on several challenging video sequences. We also motivate the need to fuse both 2D reflectance and 3D range data. In particular, the sequences (both 2D and 3D) were simulated by MIT-Lincoln Labs for several degrees of turbulence and image sizes. However, in this paper, we only present results for two levels of turbulence and image size of 64X64 pixels. We note that MIT-Lincoln Labs currently has an active 32X32 APD array in the field, but are able to simulate 32X32, 64X64, and 128X128 image sizes.

A. Robustness to Varying Views, Occlusions, and Turbulence

Let us begin with the first sequence of a truck that moves in open terrain. This is shown in Figure 6, where one can see that target occupies much of the field of view (FOV). Additionally, significant turbulence and several occlusions by trees obfuscate the object's visibility with respect to the imaging camera. Nevertheless, the algorithm successfully tracks the truck throughout the entire sequence.

The next sequence of which several frames are shown in Figure 7, demonstrates the algorithm's robustness to complete occlusions as well as a clutter environment that arises from non-cooperative objects moving around within the scene. In particular, another car that is identical in appearance (but not in depth), passes close to the target of interest. However, the algorithm successfully tracks the car throughout the sequence using the proposed tracking technique.

B. Benefits of Coupling 2D and 3D Information

In Figure 8 and Figure 9, we demonstrate inherent problems that might occur when tracking with purely 2D reflectance imagery or 3D range imagery on a challenging scenario. While similar to the sequence presented in Figure 7, here the imaging camera in this sequence loses track of the vehicle as it begins to linger near the edge of the image before visibility is completely lost. These experiments show that tracking and detection can be performed by leveraging both 2D and 3D information (as opposed to tracking with reflectance or depth alone).

For example, if non-cooperative targets such as two cars

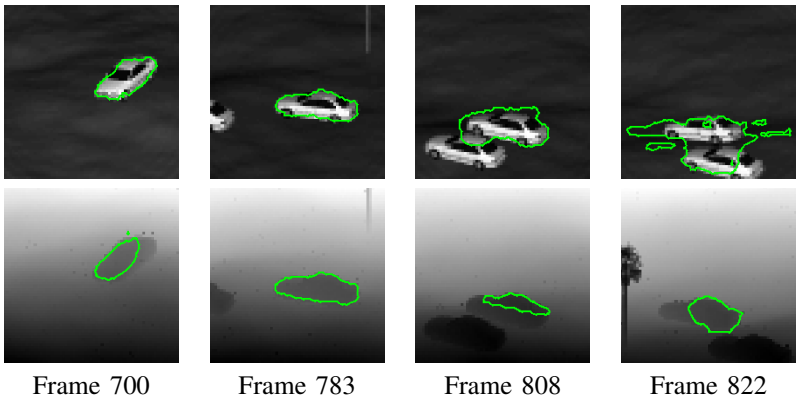


Fig. 8. Visual tracking results of a target car near a similar car. *Top Row*: Using only 2D reflectance data, the segmentation fails and leaks onto the nearby car. *Bottom Row*: Using combined reflectance and range data, the target car is tracked successfully.

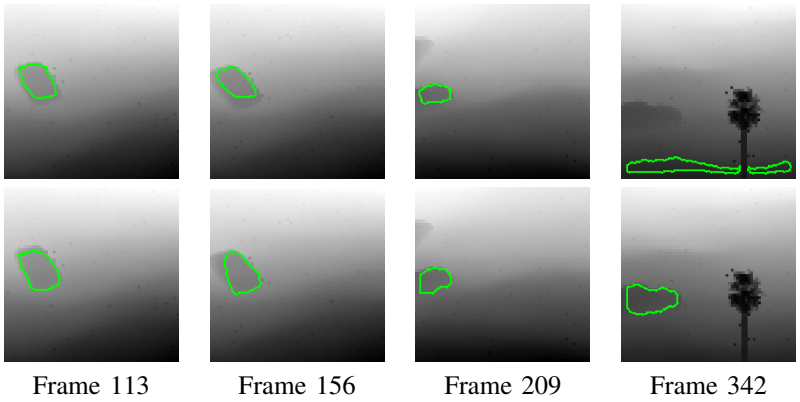


Fig. 9. Visual tracking results of a target car as it moves out of the field of view and re-enters. *Top Row*: Using only 3D range data, the detection fails and the system begins to track the background. *Bottom Row*: Using combined reflectance and range data, the target car is tracked successfully.

are present in a particular scene, it becomes increasingly difficult to distinguish the target of interest from a statistical point of view when using purely 2D reflectance. In the top row of Figure 8, we see that the active contour leaks onto the second car when the two approach each other. However, when we include 3D depth information, the target can be successfully distinguished. This is shown on the bottom row of the same figure.

Moreover, Figure 9 presents the major drawback associated with using range information alone. In the case of erratic behavior, whereby the object leaves the image view, detection becomes unreliable. That is, when the truck leaves the FOV, it will have a specific depth value, but when it re-enters its depth value may be completely different. Unfortunately, in such a case, the model for reacquisition is no longer valid and detection will fail. By leveraging on 2D information, we are able to detect and re-acquire in a more reliable fashion.

C. Quantifying Tracking Results with Ground Truth

Until now, we have provided qualitative tracking results for several difficult sequences, but have yet compared this to ground truth available by the simulator. In this section, we revisit the important notion of being able to under-segment the "target" such that pixels captured by the active contour contain relatively few or no background features at the cost of fewer target pixels. Thus, we consider the ability to capture background pixels as a "False Positive" while missing target pixels will be denoted as "False Negatives."

Several images of the sequence in which we quantify our tracking algorithm are shown in Figure 10. In particular, we would like to point out that in order for one to compare the ground truth with the tracked result as well as being able to properly register the extracted 3D point set, one must first only examine those points registered by the 3DLADAR imaging system. That is, during the pre-processing step one must artificially fill in certain regions of the image where the avalanche photo-diode array did not receive a photon count or image value. This again could be due to perhaps imaging the sky where the arrival time is infinite (not reflected). In turn, the visual images seen in this paper have been pre-processed so that they are "smooth" and continuous throughout each location in the x-y image plane. Thus, when extracting "target" features, we only extract those points that were initially registered by the imaging system. These filtered masks and points are shown in the bottom row of Figure 10.

Consequently, we are now able to compare the filtered binary masks with that of the ground truth. Interestingly, we see in Figure 11 that our False Positives are very low (below 1%) for each frame in each sequence of differing image sizes and turbulence. Again, this should be particularly

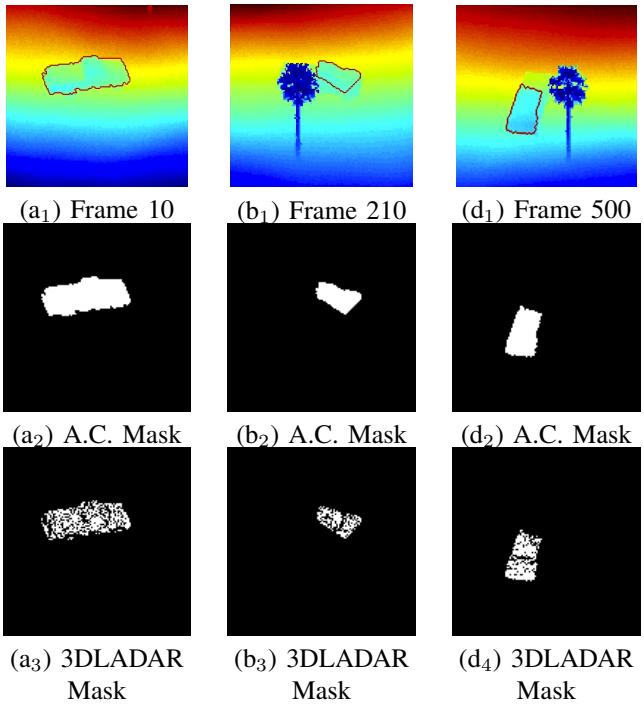


Fig. 10. Tracking Sequence of a Moving Target with Corresponding Binary Masks Provided by Active Contour and 3DLADAR Masks. Top Row: Tracked 3DLADAR Images. Middle Row: Binary Masks Associated with Active Contour (A.C.). Bottom Row: 3DLADAR Filtered Mask for Points Only Registered By System

important since we will be returning only "target" pixels and should then ease the applicability of a point set registration algorithm to estimate the target's pose.

In regards to the False Negative's, we tend to have a higher percent error when facing occlusion as shown in Figure 12. That is, we are not able to capture as much of the target as we would like to. However, we still maintain track throughout the sequence and do indeed recover from the tree occlusion as seen in Figure 10. More importantly, the target pixels that we do retain, when facing occlusions, are important features for the ICP-like algorithms. Unfortunately, this detail is not visible in the results shown. Ideally, point set registration can be performed if one is given (few) quality features of the target.

V. Conclusion

In this note, we have described a robust tracking algorithm capable of continuously tracking a target in 3DLADAR imagery despite the presence of noise, turbulence, occlusions, and temporary motion of the target out of the field of view. This is accomplished by combining geometric active contours and reasonable heuristics, which are used to re-acquire the target if tracking is disrupted. We also presented experiments to demonstrate the algorithm's robustness to turbulence, occlusions, clutter, and erratic behavior on several challenging video sequences.

References

[1] M. Albota, B. Aul, D. Fouche, R. Heinrichs, D. Kocher, R. Marino, J. Mooney, N. Newbury, M. O'Brien, B. Player, B. Willard, and J. Zayhowski. Three-Dimensional Imaging Laser Radars with Geiger-Mode Avalance Photodiode Arrays. *Lincoln Laboratory Journal*, (2):351–370, 2002.

[2] A. Betser, P. Vela, and A. Tannenbaum. Automatic tracking of flying vehicles using geodesic snakes and kalman filtering. In *Conference on Decision and Control*, pages 1649–1654, December 2004.

[3] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Calcutta Math. Soc.*, 35:99–110, 1943.

[4] A. Blake and M. Isard, editors. *Active Contours*. Springer, 1998.

[5] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *IJCV*, volume 22, pages 61–79, 1997.

[6] T. Chan and L. Vese. Active contours without edges. *IEEE TIP*, 10(2):266–277, 2001.

[7] Y. Chen, T. Huang, and Y. Rui. Parametric contour tracking using unscented kalman filter. In *Proceedings of the International Conference on Image Processing*, volume 3, pages 613–616., 2002.

[8] S. Dambreville, Y. Rathi, and A. Tannenbaum. Tracking deformable objects with unscented kalman filtering and geometric active contours. In *American Control Conference*, pages 6–13, June 2006.

[9] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi. Conformal curvature flows: From phase transitions to active vision. *Arch. Ration. Mech. Anal.*, 134(3):275–301, Sept. 1996.

[10] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Cambridge University Press, New York, NY, 2003.

[11] P.Li, T. Zhang, and B. Ma. Unscented kalman filter for visual curve tracking. *Image and Vision Computing*, 22(2):157–164., 2004.

[12] Y. Rathi, N. Vaswani, A. Tannenbaum, and A. Yezzi. Tracking deforming objects using particle filtering for geometric active contours. *IEEE transactions on pattern analysis and machine intelligence*, 29(8):1470, 2007.

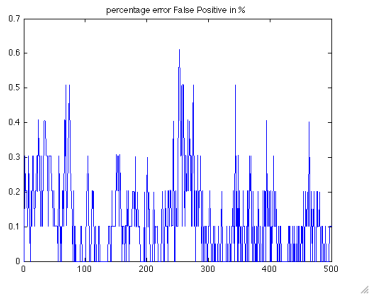
[13] R.Sandhu, S.Dambreville, and A.Tannenbaum. Point set registration via particle filtering and stochastic dynamics. *IEEE Transactions on Pattern Analysis and Machine Intelligence (Accepted - In Print)*, 2009. Accepted - In Print.

[14] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. 1999.

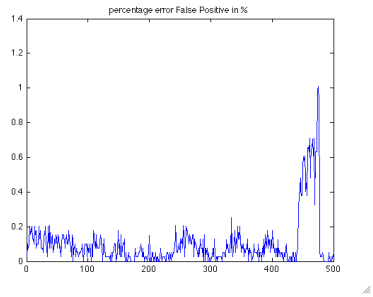
[15] D. Terzopoulos and R. Szeliski. *Tracking with Kalman Snakes*. Active Vision, MIT Press, 1992.

[16] P. Vela, M. Niethammer, G. Pryor, A. Tannenbaum, R. Butts, and D. Washburn. Knowledge-based segmentation for tracking through deep turbulence. *IEEE Transactions on Control Systems Technology*, 16(3):469–474, May 2008.

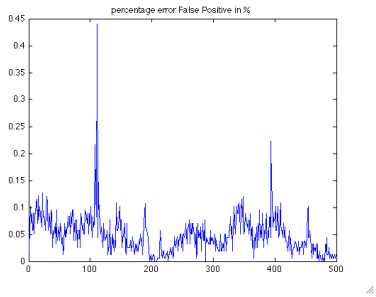
[17] A. Yezzi and S. Soatto. Deformotion: Deforming motion, shape average and the joint registration and approximation of structures in images. In *Int. Journal of Computer Vision*, volume 53, pages 153–167, 2003.



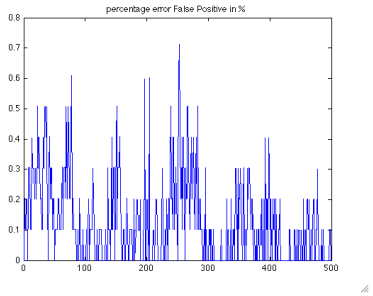
(a_1) False Positive: 32X32 No Turbulence



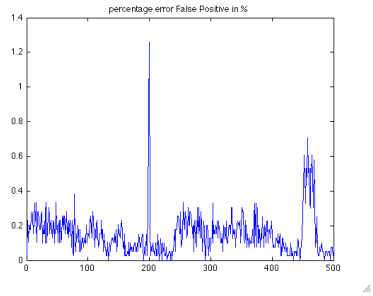
(b_1) False Positive: 64X64 No Turbulence



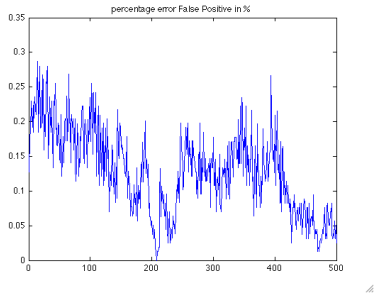
(c_1)False Positive: 128X128 No Turbulence



(a_2) False Positive: 32X32 High Turbulence

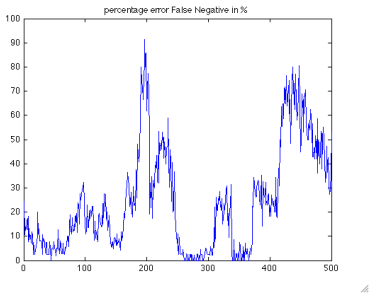


(b_2) False Positive: 64X64 High Turbulence

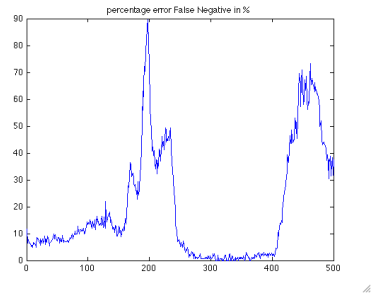


(c_2)False Positive: 128X128 High Turbulence

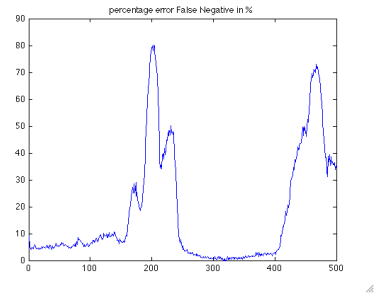
Fig. 11. Quantifying Segmentation Results with Ground Truth via False Positives. Top Row: False Positives for Image Sizes of 32X32, 64X64 and 128X128 with No Turbulence. Bottom Row: False Positives for Image Sizes of 32X32, 64X64 and 128X128 with High Turbulence. **Note: Scales are different for each image so that one can see small deviations in tracking.**



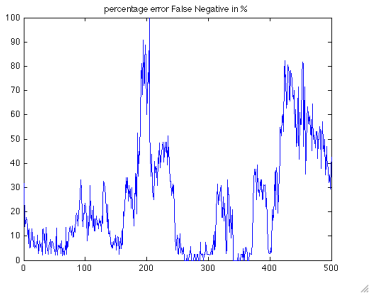
(a_1) False Negative: 32X32 No Turbulence



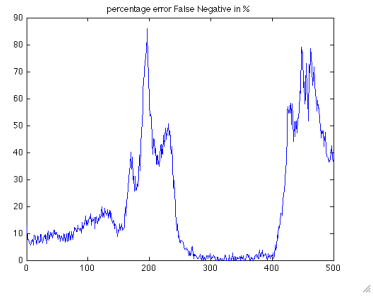
(b_1) False Negative: 64X64 No Turbulence



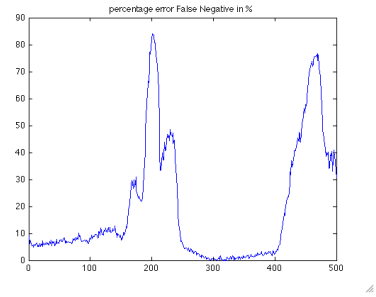
(c_1)False Negative: 128X128 No Turbulence



(a_2) False Negative: 32X32 High Turbulence



(b_2) False Negative: 64X64 High Turbulence



(c_2)False Negative: 128X128 High Turbulence

Fig. 12. Quantifying Segmentation Results with Ground Truth via False Negatives. Top Row: False Negatives for Image Sizes of 32X32, 64X64 and 128X128 with No Turbulence. Bottom Row: False Negatives for Image Sizes of 32X32, 64X64 and 128X128 with High Turbulence. **Note: Scales are different for each image so that one can see small deviations in tracking.**