A jump-diffusion algorithm for multiple target recognition using laser radar range data

Aaron D. Lanterman[a], MEMBER SPIE

[a] Coordinated Science Laboratory, Univ. of Illinois, 1308 W. Main, Urbana, IL 61801
e-mail: lanterma@ifp.uiuc.edu, web: www.ifp.uiuc.edu/~lanterma

Keywords: jump-diffusion, pattern theory, automatic target recognition, laser radar, model-based vision, Markov chain Monte Carlo

1. INTRODUCTION

Many automatic target recognition (ATR) algorithms are intimately tied to the particular sensor they are designed for, and are not readily adapted to other kinds of sensors. Grenander's pattern theory[1–3] seeks a conceptual separation between the underlying representation of a scene, the sensors used to observe that scene, and the algorithm used to perform inference using the underlying representation and the sensor model. In this paradigm, a hypothesized scene, simulated from the characteristics of the hypothesized scene elements, is compared to the collected data by a likelihood function based on sensor statistics. The likelihood is combined with prior knowledge to form a Bayesian posterior distribution. One can explore different algorithms which exploit the same underlying representation and sensor model to determine which algorithm is the most efficient. Similarly, by employing a common representation, a particular algorithm designed for one sensor may be readily modified to employ another sensor (either instead of or in addition to the original sensor) by just inserting the new sensor model.

In Ref. 4, we presented an algorithm for pattern-theoretic inference of forward-looking IR (FLIR) scenes. The logical and computational engine of the algorithm is built around a *jump-diffusion process* which provides the dynamic flexibility to accommodate scenes of differing complexity.[5–7] An ATR algorithm must have continuous and discrete aspects, since discovering the number of objects in a scene and recognizing their types is fundamentally different than deducing their orientations and positions. The "jump" and "diffusion" components respectively handle the discrete and continuous aspects of this search. The implemented jump-diffusion process may be thought of as a Markov chain Monte Carlo[8] algorithm which samples the posterior distribution.

To illustrate the generality of this framework, this communication presents results using a modification of the FLIR jump-diffusion algorithm in which a laser radar (LADAR) sensor model is substituted in place of the FLIR model.

2. PROBLEM FORMULATION

2.1. Representation of the Underlying Scene

Figure 1 illustrates a top-down representation of a particular configuration of ground-based targets. We assume the targets lie on a flat plane, so that each target can be represented by a two-dimensional

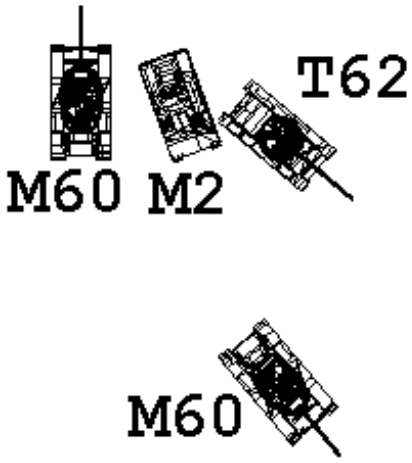

Figure 1. Top-down view of a configuration of targets.

position, a single-axis rotation, and a label indicating target type. The parameter space for an N-target scene can be written as $\mathcal{X}_N = [\Re^2 \times [0, 2\pi) \times \mathcal{A}]^N$, where $\mathcal{A}$ is a set of target types, for instance, $\mathcal{A} = \{\mathrm{M2}, \mathrm{M60}, \mathrm{T62}\}$. Since the number of targets present in the scene is not known in advance, the complete parameter space $\mathcal{X} = \cup_{N=0}^{\infty} \mathcal{X}_N$ is a union of the various N-target subspaces.

2.2. Likelihood Model for LADAR Range Data

This section describes a model for range data collected by a coherent LADAR using heterodyne detection.[9] Intensity and Doppler information, if available, can be incorporated similarly. The application of this model to pattern-theoretic ATR was first suggested in Refs. 10 and 11. Several performance-analysis studies for a single-object scenario, where the position of the object is assumed known, have been conducted by Kostakis and co-workers.[12,13] Our notation is chosen to be analogous with the notation in Sec. 3.1 of Ref. 4.

The LADAR will observe the scene through the effects of *obscuration* and *perspective projection*, in which a point (x, y, z) in 3-D space is projected onto the 2-D detector according to $(x, y, z) \mapsto (x/z, y/z)$, as shown in Fig. 2. This creates the vanishing point effect in which objects which are further away from the sensor appear closer to the center of the detector. Objects will appear skewed in different ways depending on where they appear in the image plane.

Let $\lambda(\mathbf{y})$ be the true distance from the sensor to the scene at pixel $\mathbf{y}$. We will employ the model for laser radar range data formulated by Shapiro and colleagues,[9,14,15] with a few changes to suit our purposes. In particular, we will explicitly incorporate the ambiguity inherent in coherent, heterodyne-detection LADAR range measurements, so the measurements are more accurately characterized as taking values on the torus rather than the real line. Although a wrapped Gaussian or Von Mises density would be more appropriate for toroidal data, we will employ Shapiro's original Gaussian density as a reasonable approximation. Let R_{amb} denote the length of the range ambiguity interval, i.e., ranges of r, $r + R_{amb}$, $r + 2R_{amb}$, etc. all map to the same measured range.

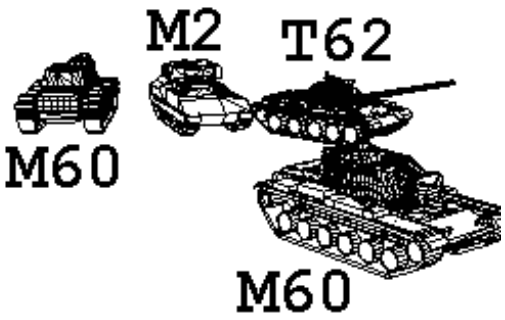

Figure 2. Wireframes of targets in Fig. 1 viewed under perspective projection and obscuration.

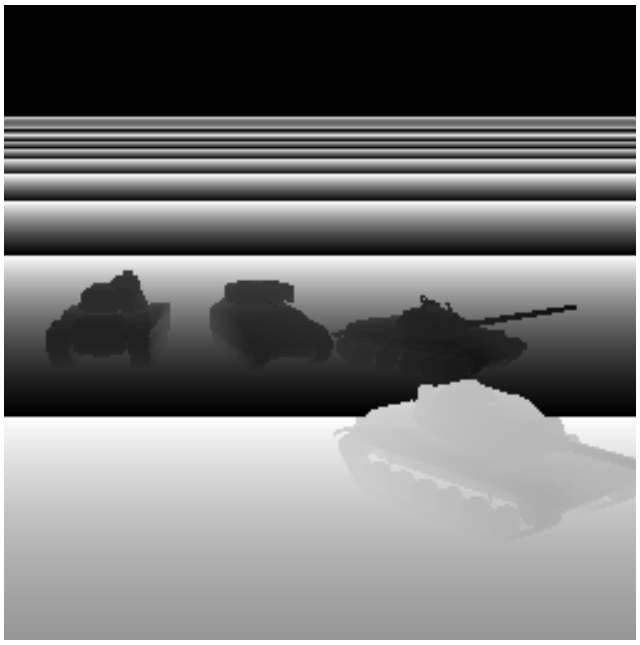

Figure 3. Noise-free LADAR range image with range ambiguity.

The loglikelihood of collecting range data $D \in [0, R_{amb}]$ given the true range image λ is

$$L_{LR}(D|\lambda) = \sum_{\mathbf{y}\in\mathcal{Y}} \ln \left\{ [1 - Pr_A(\mathbf{y})] \frac{\exp[-(D(\mathbf{y}) - \mathrm{mod}(\lambda(\mathbf{y}), R_{amb}))^2/(2\delta^2(\mathbf{y}))]}{\sqrt{2\pi\delta^2(\mathbf{y})}} + \frac{Pr_A(\mathbf{y})}{R_{amb}} \right\} \quad (1)$$

where $\delta(\mathbf{y})$ and $\mathrm{Pr}_A(\mathbf{y})$ are the local range accuracy and probability of anomalous measurement for pixel $\mathbf{y}$ given by $\delta(\mathbf{y}) = R_{res}/\sqrt{CNR(\mathbf{y})}$ and $Pr_A(\mathbf{y}) = [\ln(N) - 1/N + 0.577]/CNR(\mathbf{y})$, where $N = R_{amb}/R_{res}$, R_{res} is the range resolution, and CNR is the carrier-to-noise ratio taken to be

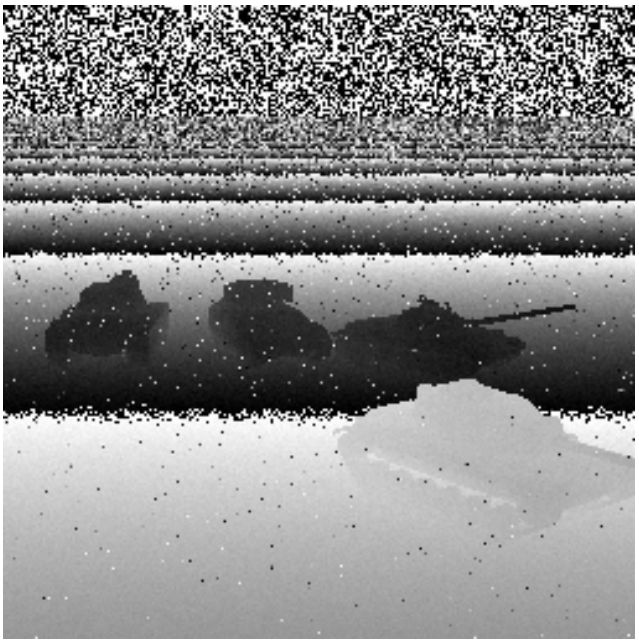

Figure 4. Sample LADAR range image with range ambiguity, anomalous pixels, and range-dependent measurement errors.

$CNR(\mathbf{y}) = \Xi \exp\left[-2\alpha\lambda(\mathbf{y})\right] / \left[\lambda(\mathbf{y})\right]^2$, where α is the atmospheric extinction coefficient and Ξ is a constant derived from the properties of the laser radar.[9,14,15] Notice that the probability of anomaly $Pr_A(\mathbf{y})$ and the local range accuracy $\delta(\mathbf{y})$ increase with hypothesized distance $\lambda(\mathbf{y})$.

Figure 3 presents an example of a noiseless range image (with ambiguity) corresponding to Fig. 2. A sample of this image corrupted with anomalous pixels and range-dependent measurement error is shown in Fig. 4. The parameters have been chosen to illustrate overall effects and are not intended to be correspond to any particular real LADAR system or tactical scenario. In particular, they were chosen to emphasize the range-dependent errors; notice that the pixels further from the detector are noisier than those near the detector.

Given an estimated configuration x and a collected image D, we can write the loglikelihood parameterized by the hypothesized underlying scene as

$$L(D|x) = L_{LR}(D|render(x)), \tag{2}$$

where $render(x)$ represents the operation of rendering, via perspective projection and obscuration, the range to elements of the scene. Note that sensor fusion is naturally achieved in this framework by adding the loglikelihoods for different sensors. Similarly, if several images are collected over time, one can add the loglikelihoods for the different frames. The $render(x)$ function will be different for each frame due to the motion of the sensor platform.

Here, we assume a uniform prior on target types, positions, and orientations, so the posterior is proportional to the likelihood.

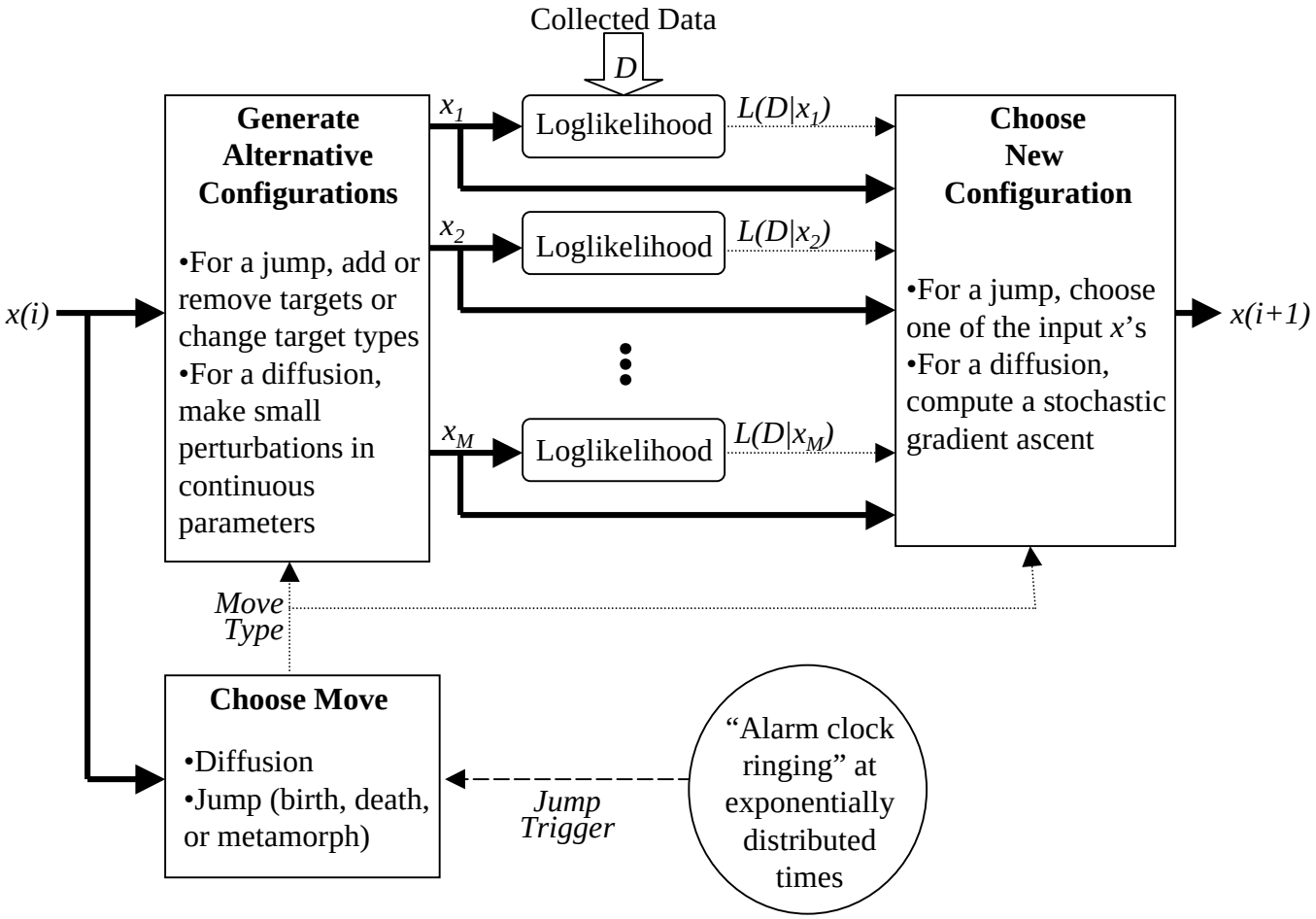

Figure 5. Block diagram illustration the overall operation of the jump-diffusion ATR algorithm.

2.3. Implementation using Z-Buffer Algorithms

Many computer graphics engines, including Silicon Graphics hardware, account for obscuration with a *z-buffer algorithm.* As object pixels are rendered, their distance is stored in a pixel-registered z-buffer. New pixels are only written to the rendered image if their distance is less than what is currently stored in the z-buffer. After the scene is fully rendered, the final contents of the z-buffer provide ranges as a by-product of the rendering algorithm. This facilitates the rapid simulation of LADAR scenes.

3. EXAMPLE

We have modified our Silicon Graphics implementation of the jump-diffusion algorithm for FLIR[4] to employ LADAR data instead. To create our LADAR ATR algorithm, we simply substituted the LADAR loglikelihood calculation in place of the FLIR loglikelihood calculation. The majority of the jump-diffusion code remained unchanged.

We refer the reader to Ref. 4 for extensive details of the original FLIR ATR algorithm. Here we merely present a block diagram (Fig. 5) to provide an intuitive feel for how it operates. The

algorithm produces a series of estimates $x(i)$. Here, the time i refers to "algorithmic time" or "process time" which should not be confused with the real time over which the data is gathered. We initialize the algorithm by setting $x(0)$ to be an empty configuration. At each iteration, a number of alternative hypotheses, denoted $x_1, x_2, \ldots x_M$, are generated from the current hypothesis $x(i)$ according to a chosen "move type." The loglikelihood of each alternative hypothesis is computed.

The most common move type is a diffusion; in this case, the alternative hypotheses represent small perturbations in continuous parameters. These are used to compute numerical derivatives of the loglikelihood. A new estimate $x(i+1)$ is computed by following the gradient of the loglikelihood along the continuous parameters. A small random noise term is added to the result of the gradient update.

At exponentially distributed times, a jump move is performed instead. One of three species of jump moves is selected: a birth, a death, or a "metamorph." For a birth move, the alternative hypotheses represent the addition of a single new target, with the remaining targets left unchanged. This allows the algorithm to discover new targets. For a death move, the alternative hypotheses each have a single target removed. This permits the algorithm to remove unlikely hypothesized targets. A metamorph move consists of trying different target types (M2, T62, M60, etc.) for a particular target (including large-scale changes in orientation), allowing the algorithm to "change its mind" about the type of a target. For each kind of jump move, one of the alternatives is chosen (or the original hypothesis $x(i)$ is kept) probabilistically based on the loglikelihood. Hypotheses with greater loglikelihood have a greater chance of being chosen. Between jumps, the diffusions refine estimates of position and orientation.

The computational complexity per iteration depends on the kind of move chosen at a particular iteration. For a diffusion move, the system must simulate a scene and compute a loglikelihood (the sum in Eq. 1) twice (in order to compute a numerical derivative) for each parameter for each target. For an N target scene, this would require $6N$ renderings and likelihood computations (since there are three parameters: two position parameters and one orientation parameter). Similar analysis can be done for the jump moves; see Ref. 4 for more details. The distance rendering is quite fast using dedicated graphics hardware as described in Sec. 2.3. The primary bottleneck is the likelihood computation. In our current implementation, for each likelihood computation, we compute the likelihood over all pixels in the image. This is somewhat redundant, since a small change in the scene parameters will only change a small number of pixels on the image. Hence, it could be made faster by only computing the likelihood for the pixels which change from one likelihood computation to the next. Such optimizations are a good subject for future work.

Interesting snapshots of a sample path analyzing the data in Fig. 4 are shown in Fig. 6. The current hypothesis is shown as a white outline. The algorithm first finds the M60 on the right. Since it is closer to the detector, it takes up more pixels, and the algorithm chooses it since it can "explain" a large portion of the data. In the third iteration, a birth move discovers the M60 on the left, although it gets the orientation wrong. By iteration 11, the diffusions have pulled the M60 on the right to the correct position. In iteration 12, a "metamorph" move makes a large-scale orientation change to correct the orientation of the M2 on the left. In iteration 24, a birth move finds the M2; also note that the diffusions have refined the position of the leftmost M60. Iteration 32 shows the algorithm birthing an M2 (facing the wrong direction) over the T62. The algorithm changes its mind and switches this incorrect M2 to a T62 (now facing the correct direction) via a

metamorph move in iteration 38. By iteration 130, the diffusions have refined the pose of the T62. Notice that, in this preliminary experiment, the algorithm incorrectly estimates the orientation of the M2.

4. CONCLUSIONS

This correspondence considered a pattern-theoretic framework for ATR which may be readily adapted to employ different kinds of sensors. To illustrate its flexibility, a jump-diffusion algorithm originally designed for ATR using FLIR data was adapted to employ LADAR range data. It makes full use of the statistics of the available sensors and seeks optimal Bayesian solutions, and hence has the potential for better performance than suboptimal algorithms that do not incorporate this statistical knowledge. Further studies using, for instance, Monte Carlo simulation, would be needed to confirm and quantify this improvement by comparing the presented approach against a variety of other algorithms under a wide variety of noise conditions. However, our present interest in the pattern-theoretic approach is only partially motivated by improvements for this particular LADAR sensor; it is primarily motivated by the ease and elegance with which additional, quite different, sensors can be fused into the resulting ATR system.

Our algorithm assumed that the position of the platform is known relative to ground using, for instance, GPS. Unfortunately, at the time of this study, we did not have access to real LADAR data which had this kind of auxillary information. (An alternative to using GPS data would be to make the platform position itself a parameter of interest; however, this would complicate matters, and we wanted this exposition to be as simple as possible.) Of course, a vital next step will be to test the algorithm using real LADAR data.

ACKNOWLEDGMENTS

The author would like to thank his graduate advisors, Profs. Michael Miller and Donald Snyder, for guidance over the course of this research. This research was conducted while the author was with the Department of Electrical Engineering at Washington University in St. Louis and was supported by ARO DAAH04-95-0494, ONR N00014-94-1-0859, ONR/AASERT N00014-94-1-1135, ONR N00014-95-0095, and ARO/AASERT DAAH04-94-G-0209. He is currently supported by DARPA Contract F49620-98-1-0498.

REFERENCES

1. U. Grenander, *Elements of Pattern Theory*, Johns Hopkins Univ. Press, 1996.
2. U. Grenander, *General Pattern Theory: A Mathematical Study of Regular Structures*, Oxford Univ. Press, 1994.
3. U. Grenander and D. Keenan, "Towards automated image understanding," *Journal of Applied Probability* **16**(2), pp. 207–221, 1989.
4. A. Lanterman, M. Miller, and D. Snyder, "General Metropolis-Hastings jump diffusions for automatic target recognition in infrared scenes," *Optical Engineering* **36**, pp. 1123–1137, April 1997.
5. U. Grenander and M. I. Miller, "Representations of knowledge in complex systems," *Journal of the Royal Statistical Society B* **56**(3), pp. 549–603, 1994.

6. M. Miller, A. Srivastava, and U. Grenander, "Conditional-mean estimation via jump-diffusion processes in multiple target tracking/recognition," *IEEE Trans. on Signal Processing* **43**, pp. 2678–2690, November 1995.
7. A. Srivastava, M. Miller, and U. Grenander, "Multiple target direction of arrival tracking," *IEEE Trans. on Signal Processing* **43**, pp. 1282–1285, May 1995.
8. D. S. W. Gilks, S. Richardson, ed., *Practical Markov Chain Monte Carlo*, Chapman and Hall, 1996.
9. J. Shapiro, B. Capron, and R. Harney, "Imaging and target detection with a heterodyne-reception optical radar," *Applied Optics* **20**, pp. 3292–3313, October 1981.
10. A. Lanterman, M. Miller, and D. Snyder, "The unification of detection, tracking, and recognition for millimeter wave and infrared sensors," in *Radar/Ladar Processing*, W. Miceli, ed., vol. SPIE Proc. 2562, pp. 150–161, (San Diego, CA), August 1995.
11. A. Lanterman, M. Miller, and D. Snyder, "Automatic target recognition via the simulation of infrared scenes," in *Proc. of the 6th Annual Ground Target Modeling and Validation Conf.*, pp. 195–203, U.S. Army TACOM, (Houghton, MI), August 1995.
12. J. Kostakis, M. Cooper, T. Green, M. Miller, J. O'Sullivan, J. Shapiro, and D. Snyder, "Multispectral active-passive sensor fusion for ground-based target orientation estimation," in *Automatic Target Recognition VIII*, F. Sadjadi, ed., vol. SPIE Proc. 3371, pp. 500–507, (Orlando, FL), April 1998.
13. J. Kostakis, M. Cooper, T. Green, M. Miller, J. O'Sullivan, J. Shapiro, and D. Snyder, "Multispectral sensor fusion for ground-based target orientation estimation: FLIR, LADAR, HRR," in *Automatic Target Recognition IX*, F. Sadjadi, ed., vol. SPIE Proc. 3718, pp. 14–24, (Orlando, FL), April 1999.
14. T. J. Green and J. H. Shapiro, "Detecting objects in 3D laser radar range images," *Optical Engineering* **33**, pp. 865–873, March 1994.
15. T. J. Green and J. H. Shapiro, "Maximum-likelihood laser radar range profiling with the expectation-maximization algorithm," *Opt. Eng.* **31**, pp. 2343–2354, 1992.

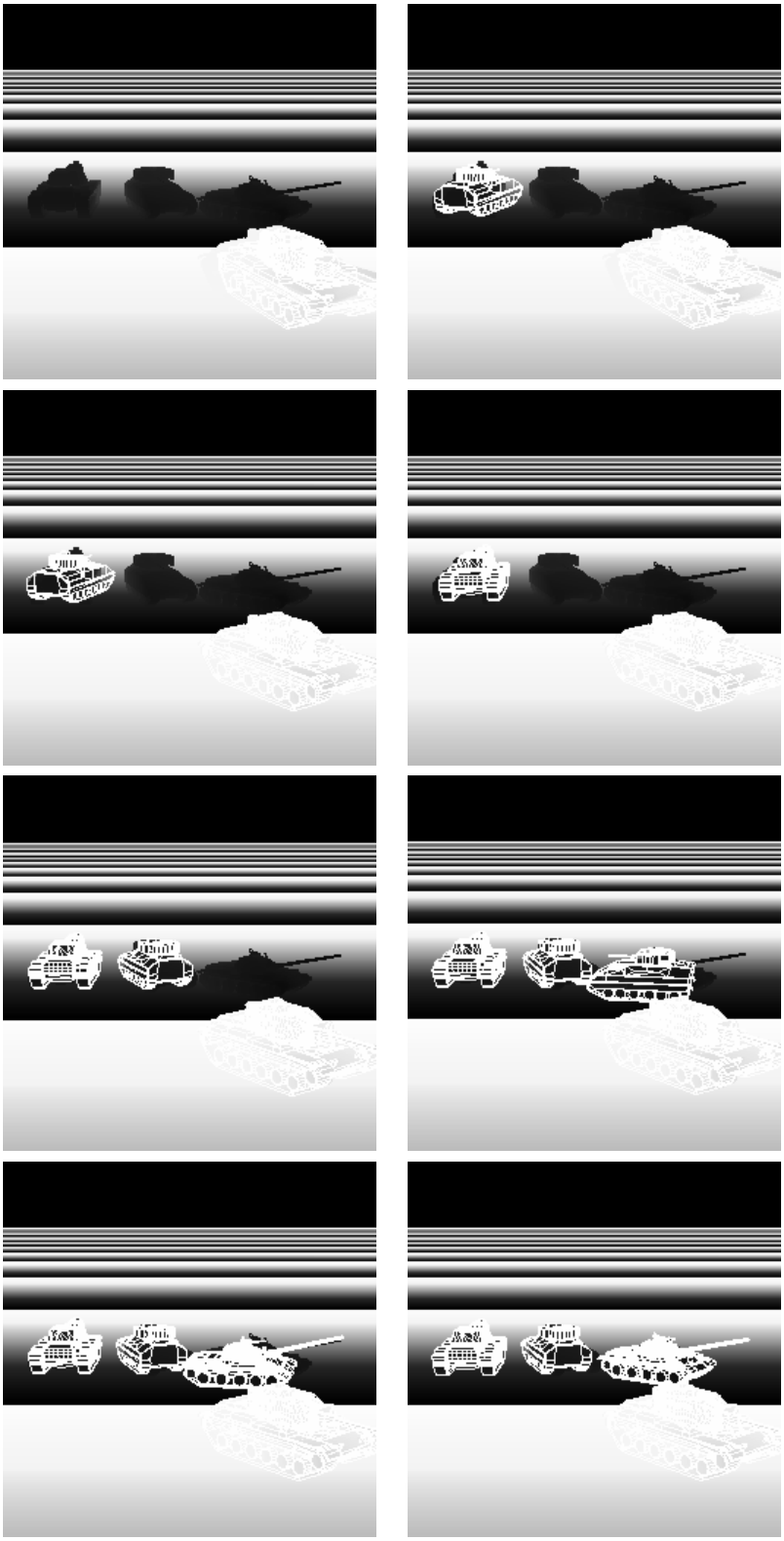

Figure 6. Iterations 1, 3, 11, 12, 24, 32, 38, and 130 of a jump-diffusion process for LADAR range data.