PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

Visual analysis for live LIDAR battlefield change detection

Thomas Butkiewicz, Remco Chang, Zachary Wartell, William Ribarsky

SPIE.

Visual analysis for live LIDAR battlefield change detection

Thomas Butkiewicz, Remco Chang, Zachary Wartell, and William Ribarsky
University of North Carolina at Charlotte

ABSTRACT

We present the framework for a battlefield change detection system that allows military analysts to coordinate and utilize live collection of airborne LIDAR range data in a highly interactive visual interface. The system consists of three major components: The adaptive and self-maintaining model of the battlefield selectively incorporates the minority of new data it deems significant, while discarding the redundant majority. The interactive interface presents the analyst with only the minute portion of the data the system deems relevant, provides tools to facilitate the decision making process, and adjusts its behavior to reflect the analyst's objectives. Finally, the cycle is completed by the generation of a goal map for the LIDAR collection hardware that instructs as to which areas should be sampled next in order to best advance the change detection task. All together, the system empowers analysts with the ability to make sense of a deluge of measurements by extracting the salient features and continually refining its definitions of relevancy.

Keywords: LIDAR, change detection, dynamic terrain, UAV

1. INTRODUCTION

Modern airborne LIDAR acquisition systems are able to collect range measurements of the ground at rates approaching hundreds of thousands of points per second. Typically, after a collection flight has completed, the sampled points are then processed en masse into a terrain model for final use by the client. However, with the introduction of LIDAR acquisition hardware small enough to deploy on multiple military unmanned aerial vehicles (UAVs), it is particularly attractive to be able to process and use the collected point samples as a live data stream. When sampling environments prone to unpredictable changes, our chronologically continuous modeling structure is superior to discrete time steps. This approach is especially apt for battlefield use, where detecting and presenting changes as they occur is of supreme significance for enhancing situational awareness.

As the system receives points from LIDAR acquisition hardware, it compares these points against the existing battlefield model. If an incoming point merely reflects the current model (redundant data) it is discarded and the model notes that the measurement confirmed that particular area of the model as being satisfactorily representative of the physical terrain. An incoming point that deviates significantly (more so than errors could account for) from the existing model is evaluated to determine if it is a legitimate change in the physical terrain (such as a roadblock constructed). Changed points are passed to the change aggregator/tracker which builds and maintains models of changes over time. These change models are the primary focus of the task and application, and can be passed through an automatic target recognition algorithm. If a point is reported as a potential change in an area where the model is incomplete, we can refine the model to include that measurement. As more data is obtained for that region, we can make the decision as to whether we permanently accept the model refinement, or report it as being a temporary change.

Analysts do not simply observe this data collection process. Instead, through the use of a highly interactive visual interface, they can continuously make assessments concerning the value and significance of changes as the system reports them. Those changes deemed relevant can be output to end-users, in this case soldiers in the field with PDAs, or used to update live battlefield models. To promote future change discoveries, the system uses goal maps to coordinate the ongoing acquisition efforts, concentrating new data collection on areas that have shown to be active, while being careful not to allow the awareness of less active regions to stagnate. The analyst can leverage their judgment with tools such as regions-of-interest to guide the data collection progress beyond the limited abilities of any automatic algorithm.

Defense and Security 2008: Special Sessions on Food Safety, Visual Analytics, Resource Restricted Embedded and Sensor Networks, and 3D Imaging and Display, M. S. Kim, K. Chao, W. J. Tolone, W. Ribarsky, S. I. Balandin, B. Javidi, S.-I Tu, Eds., Proc. of SPIE Vol. 6983, 69830B, (2008) · 0277-786X/08/$18 · doi: 10.1117/12.777158

2. PREVIOUS WORK

A number of LIDAR visualization and analysis tools are available as stand-alone packages as well as add-ons to CAD systems[1]. Each tool has its strength and weakness in terms of visualization, measurement, classification, segmentation, surface fitting, etc. Some tools are self-contained but somewhat limited such as Quick Terrain Modeler[2]. Others are self-contained, multi-component terrain analysis tools such as Mars Viewer[3] and Polyworks[4]. Several are either integrated in a larger CAD software package, such as Terrasolid[5] which integrates with Bently's Microstation CAD software, or support extensibility through a proprietary programming interfaces such as Fledermaus[6]. Finally, a number of terrain analysis modules are also available for engineering programming and analysis tools such as Matlab[7]. The visualization and analysis components of these tools could all serve various niches in the framework we present in this paper. Those whose functionality is programmatically extensible would be the best candidates. This is particularly important since our presented framework is organized around distributed computing.

The CUBE algorithm by Brian Calder[8] and its integration into the workflow of hydrographic surveyors[9] is of particular relevance to our framework. Modern hydrographic surveyors do sonar bathymetry surveys from marine vessels. This process gathers large volumes of data and ideally the scans are assembled to some degree in real-time and allow for vessel course adjustments for re-scannings. The CUBE algorithm uses a sophisticated model of multi-beam sonar instrumentation error and integrates an optimal Bayesian estimator. The error is carried into the terrain model and CUBE maintains multiple hypotheses for depth values and uncertainty measures. The CUBE algorithm is integrated into an interactive 3D visualization. The algorithm highlights depth estimates with high uncertainty and allows the user to interactively examine the surface and the errors at these locations. The user can interactively select the most appropriate depth values for these uncertain cases. The system maintains both 'daily' and a 'cumulative' model of the mapped surface and both are updated as new data is either automatically generated or manually adjusted. The complete system integrates the Fledermaus interactive visualization tools. This working system embodies a number of concepts in the framework we present. Our framework, however, is applied to LIDAR and is significantly broader in scope. It includes a distributed computing design, target recognition and a significant focus on change detection and terrain model maintenance. Further, the temporal component of our framework assumes continuous data updates from multiple LIDAR data sources.

3. SYSTEM DESIGN

The system consists of seven separate processing components each with their own tasks to transform, modify, and populate the stream of data throughout. The data flow in and out of each of these components is illustrated in Figure 1. The raw data is first collected by airborne LIDAR sensors, which transmit their measurements to the data stream collector and distributor. Here the data is split into chunks of a manageable size and fed to multiple change detection processors. This type of parallel, distributed processing is necessary, as the change detection process is very calculation-intensive and must be done in real time for a massive stream of incoming data. Detected change points are then aggregated into change models which are tracked over time. After attempting automatic target recognition, these models are delivered to the interactive application for analyst attention. Finally, a goal map is maintained that directs the collection efforts of the airborne sensors.

3.1 Source data and assumptions

The bare minimum for incoming data required for this system would be a single stream of geo-referenced X,Y,Z point locations. Additional information, such as return classification data (differentiating between soil, vegetation, rooftops, etc) could be used to augment and strengthen the change detection and model building algorithms; however it shall not be required for basic functionality. By maintaining a low requirement on input complexity, one may be able to utilize less-sophisticated hardware, a likely advantage due to size and weight constraints involved with deploying sensors on smaller UAV platforms.

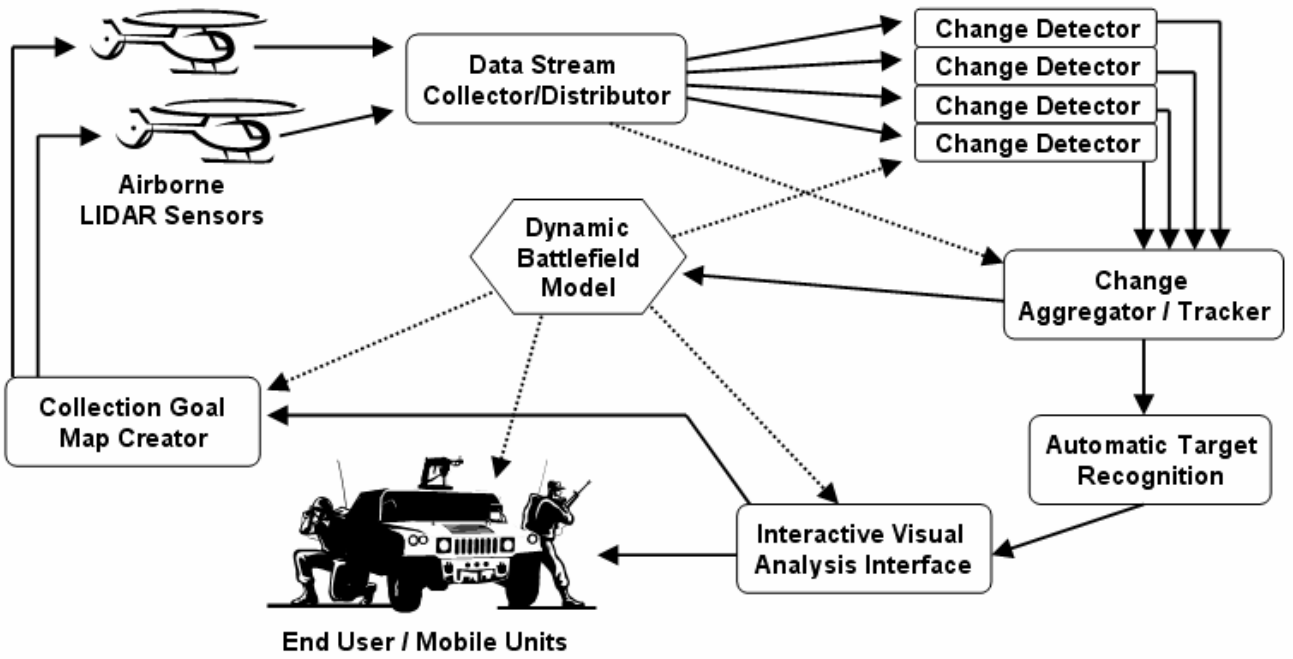

Fig. 1. Flowchart showing the cycle of data flow in the system. The airborne LIDAR sensors provide data to the data stream collector, which distributes it to multiple change detection processors. These report any points that have changed to the change tracker, which extracts larger changes by grouping change points into larger models/structures. These can be fed through an automatic target recognition process and then to the interactive application. The interactive application presents the analyst with the extracted (and possibly identified) changes. The analyst can then inspect these changes, and indicate which are significant and thus are to be exported as updates to the end users, in this case mobile units on the battlefield. The interface can also be used to manually affect the goal map creation process, which completes the data cycle by sending flight commands to the collection hardware.

3.2 Initial model creation

While not absolutely required, it is assumed that, at the start, an initial model can be built either from accessible GIS datasets or previous LIDAR scans of the area to be monitored. Standard GIS data such as digital elevation maps (DEMs), building footprints (often compiled for tax or insurance purposes), and aerial/satellite photography can all be used to create substantially relevant initial models. For example see reconstruction efforts in Suveg et al.[10] If previously acquired LIDAR scans are available, they can provide all of the necessary data required to create the ideal initial model for this system.

Areas of discontinuous coverage in an initial model will be filled in as part of the normal behavior of the systems main loop. This includes the case in which there is no initial model provided.

3.3 Data collection

Once the initial model has been built, the main loop of data collection can begin. As the airborne collection hardware samples the surface, new points are continuously delivered to the change detection algorithm. Each airborne sensor provides its own stream of measurements into the system. Because the change detection algorithm does not rely on the incoming samples being in any particular order, or even in bursts from the same stream, the incoming point locations from all streams can be merged into a single stream.

Finally, due to the highly parallelizable nature of the change detection algorithm, the combined stream of incoming data can then be split into smaller chunks such that all processing units tasked with change detection duty receive manageable amounts of data. In this way, by distributing the incoming data to multiple change detection processing units, the system can scale to accommodate as many airborne sensor units as needed in order to maintain suitable coverage and redundancy, while preserving the real-time behavior of the system.

3.4 Change detection

Each change detection process first requires a connection to the main battlefield model. Simply put, the change detection algorithm compares incoming points against the existing model, ignores those that conform to the model, and forwards those that do not conform to the change tracker/aggregator component. There are various methods that can be utilized to evaluate the candidacy on incoming points. Briefly presented here is the existing technique[20] that we have previously developed and used with success to extract changes to a urban environment from annual LIDAR scans. It is attractive because it is both highly parallelizable and its decision making algorithm considers the error present in the collection hardware and model creation process.

An incoming point is first projected onto the existing model to determine the local region to which the point corresponds. The surrounding data points from this region are then extracted from the model, and based on their known accuracy (source dependent) and density (a highly sampled area is more reliable than a sparsely sampled area), the range of heights of the surface at that point is then predicted. The incoming point is then compared to this range; if it significantly exceeds it (defined by user specified standard deviation or hard limits), then the deviation of the point is assumed to be due to a change in the sampled environment and forwarded to the change tracker for further processing.

An example of the output of the change detection component is show in Figure 2. There you can see the marked change points in red. Notice that there are many marked points scattered about due to the low resolution of the LIDAR data in comparison to the objects being sampled, as well as vegetation growth. In Figure 3, you can see that after change aggregation (detailed in the next section) only the marked points forming significant objects remain.

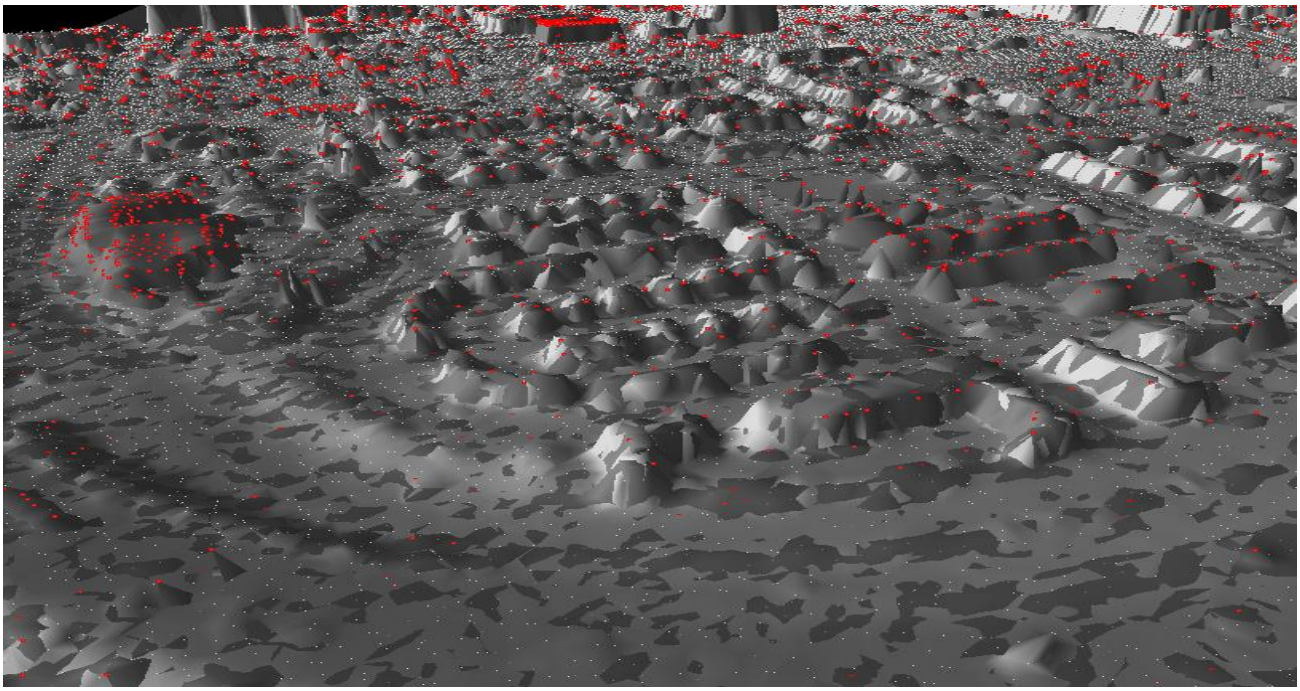

Fig. 2. Example output from the change detection process. The lighter gray patches are the initial model's surface, while the darker gray patches are the subsequent scan's surface. Points in white have been deemed to be conforming by the change detection algorithm, while those highlighted in red have been marked as deviating due to external environmental changes (as opposed to internal shifts due to errors and uncertainty). The majority of the marked points are due to the low resolution of the initial scan and vegetation, however a significant number correspond to five new buildings, which remain after the change aggregation process as shown in Figure 3.

3.5 Change Aggregation and Tracking

As new change points arrive from change detection processors, this component is primarily tasked with grouping together nearby points to form change models. These change models are then tracked over time to determine if they are still developing (new points are coming in and adding detail) or established (area has been scanned again showing no further changes). Change models are relayed to the interactive application, automatic target recognition, and

incorporated into the main battlefield model as appropriate. See Figure 3 for an example of the change models formed and output from the input change points shown in Figure 2.

When a change point first arrives from a change detection processor, its location within or distance to any existing change models being tracked is determined. Points not falling within range of existing models become new models themselves. However, if the new point is within the specified range of an existing model, we must test whether inserting that point into the change model would meaningfully increase its detail. An effective method of determining this is to utilize the vertex removal cost calculation from a mesh simplification algorithm in reverse to determine instead the benefit of adding a vertex to the model. QSlim[11] provides an excellent metric known as "quadric error" that quantifies the geometric cost of edge contraction (which removes a vertex). By inserting the point to be tested into the model, then calculating the quadric error of removing that point, we can determine exactly how much it would contribute to the detail of the model. If the detail it would contribute is beyond a specified amount, we inset it permanently into the change model and note that the change model is currently in the "developing" state. When a developing change model exceeds a specified minimum size or complexity, it is considered to be a significant change model, and each time it is updated with new points, the model is sent to the interactive application as a urgent developing event. Conversely, if inserting the new point into the model would not yield any significant new detail, we discard the point and note that the change model is progressing from the "developing" state to the "established" state. This can be accomplished by incrementing a counter, such that after a specified number of redundant points are detected (and time passed), the model is considered to be established. When a change model is identified as established, it is sent to the main battlefield model to be incorporated into the permanent state of the overall model. If incoming points corresponding to an existing change model show that the area has now reverted back to match the initial battlefield model, we alert the interactive application as to the change's disappearance, and discard the change model.

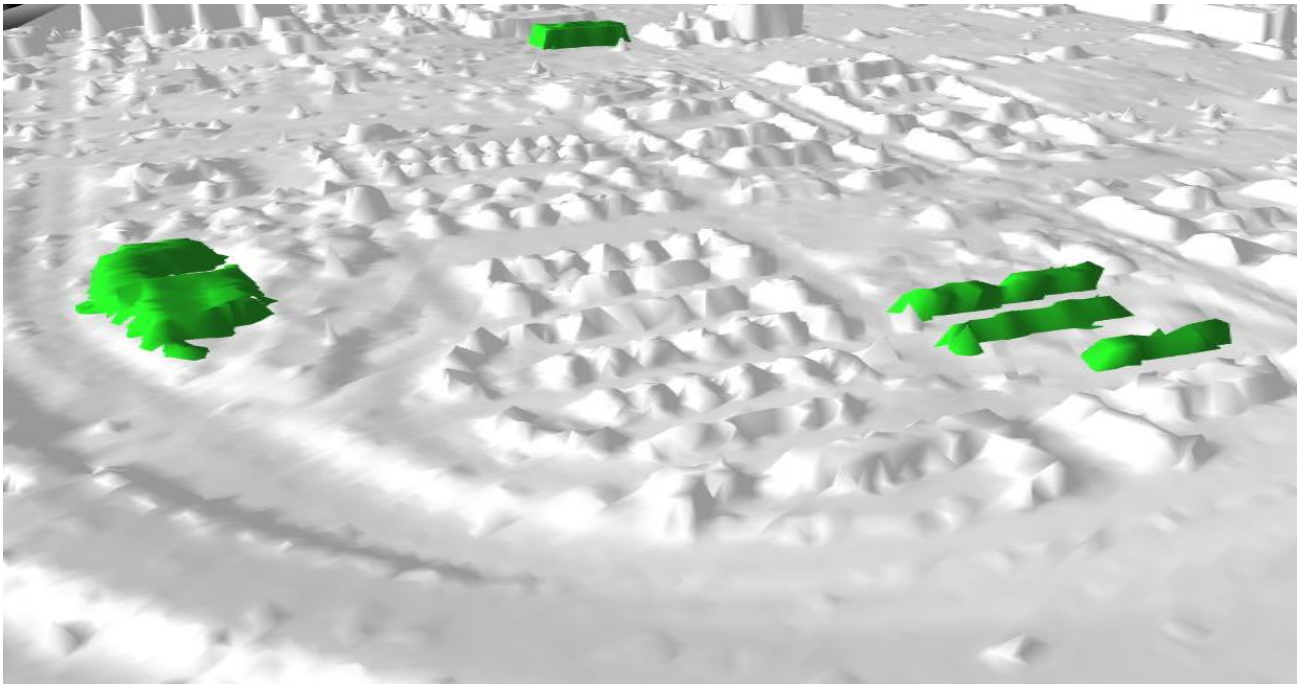

Fig. 3. Example output from the change aggregation process. Compare to the cluttered collection of marked points used as input (red points) in Figure 2. The change points not forming significant changes are suppressed, while those forming significant change, here new buildings, are aggregated into change models for tracking and output.

An example scenario consists of a truck being moved to produce a roadblock in an alleyway. On the first LIDAR pass, the first half of the truck is sampled. A change detection processor sends the first point to the tracker, which finds no existing changes in that area, and creates a new model to track changes there. As the other points from that pass are

received from the change detectors, they are evaluated and prove to add detail to the change model, which begins to take shape. At this point the analyst using the interactive application has been informed of an emerging change being detected, and can see the front half of the truck. On the second LIDAR pass, the remainder of the truck is scanned and detected. The new points add significant detail to the change model and the analyst is alerted to the updated visual. Points from subsequent LIDAR passes do not yield more detail and thus the truck has been stationary. Eventually the change model is sent to the main battlefield model to be incorporated. If a later pass revels the truck has moved, a new change model will be created to show its absence, and the cycle will repeat to maintain the accuracy of the model.

3.6 Automatic Target Recognition

To further strengthen the visualization provided to the analyst in the interactive application, it is advantageous to insert an automatic target recognition (ATR) processor between the change aggregator/tracker and the interactive application. By having the change tracker send the change models through the ATR processor, additional information can be provided to supplement the purely visual nature of the change model itself. This additional information could take a nominal form such as "Flatbed Truck", "Station Wagon", or "Armored Personnel Carrier" or a predictive form, in which it detects that an incomplete change model matches a portion of a larger known model type. For example, in our truck/road block example from section 3.5, when the samples from the first half of the truck have been modeled, the ATR may be able to determine that it is indeed the front half of a flatbed truck, and not only report the type of object it believes it has detected, but could also draw the predicted second half of the truck before it is actually sampled by a LIDAR sensor.

The necessary techniques for this type of ATR from LIDAR data have already been heavily investigated by several researchers. For example, Vasile and Marino[12] present a successful method for LIDAR based ATR on military type targets (including under vegetative cover and camouflage). Funkhouser et al.[13] have developed a search engine capable of identifying and returning existing models in its extensive collection based on input in the form of hand drawn pictures from any perspective.

3.7 Interactive Application

The interactive application, as opposed to the other components which are largely automated aside from user specified thresholds, is centered on providing the analyst with a means to not only observe the change detection process, but monitor, guide, and influence the actions of the entire system.

The behavior of the system thus far has been to reduce the torrential streams of sensor data, through an array of processors, into a manageable, easily observable stream of events. These events, received from the change tracker, are alerts pertaining to newly detected changes, previously detected changes acquiring more detail, moving, or disappearing. It is here in the interactive application that these events are visually presented to the analyst. The analysts task is primarily to use superior human object and threat identification skills to determine which changes are the most relevant to the end users in the field. As such, the highest priority event is a new change being detected, requiring urgent evaluation by the analyst. Such an event would be displayed by depicting the change model in its current state against the existing battlefield model, along with any additional information from the automatic target recognition system. A visual (and perhaps audible) alert should draw immediate attention to its appearance, to overcome issues of change blindness.

Once the analyst is alerted to the new change, he can proceed to indicate the course of action the tracking system should take. He can mark the change as not being significant, in which case the tracker will no longer send update alerts unless there are radical changes. This will also signal the change tracker to immediately update the battlefield model with the change model. Alternatively, the analyst can mark the change as being important and thus automatically forward its location and visual model to the end users in the field. This action can also cause the change tracker to send additional update events at a more frequent rate as well as affect the priority of resampling the local region through the goal mapping component (detailed in the next section). Changes that are unidentifiable can be marked as needing more detail, in which case the behavior (increased resampling priority and frequency of updates) is the same as marking as important, with the exception that no data passes to the end users until final identification has been made.

For changes marked important or as needing more detail, the visual tracking component seeks to link the models in the visualization with other previously detected models of similar shape and structure that are located nearby. For example if a change in the shape of a vehicle is detected at time t, then a similar shape is detected nearby at time $t+1$, the change models can reasonably be assumed to be related due to their spatial and temporal proximity. These changes can then be

visually connected, over time forming a path tracking the movement of the object. This path can then be relayed to the end users in the field as well, giving them both knowledge of an approaching threat and a perhaps even a predicted path of travel based on street data.

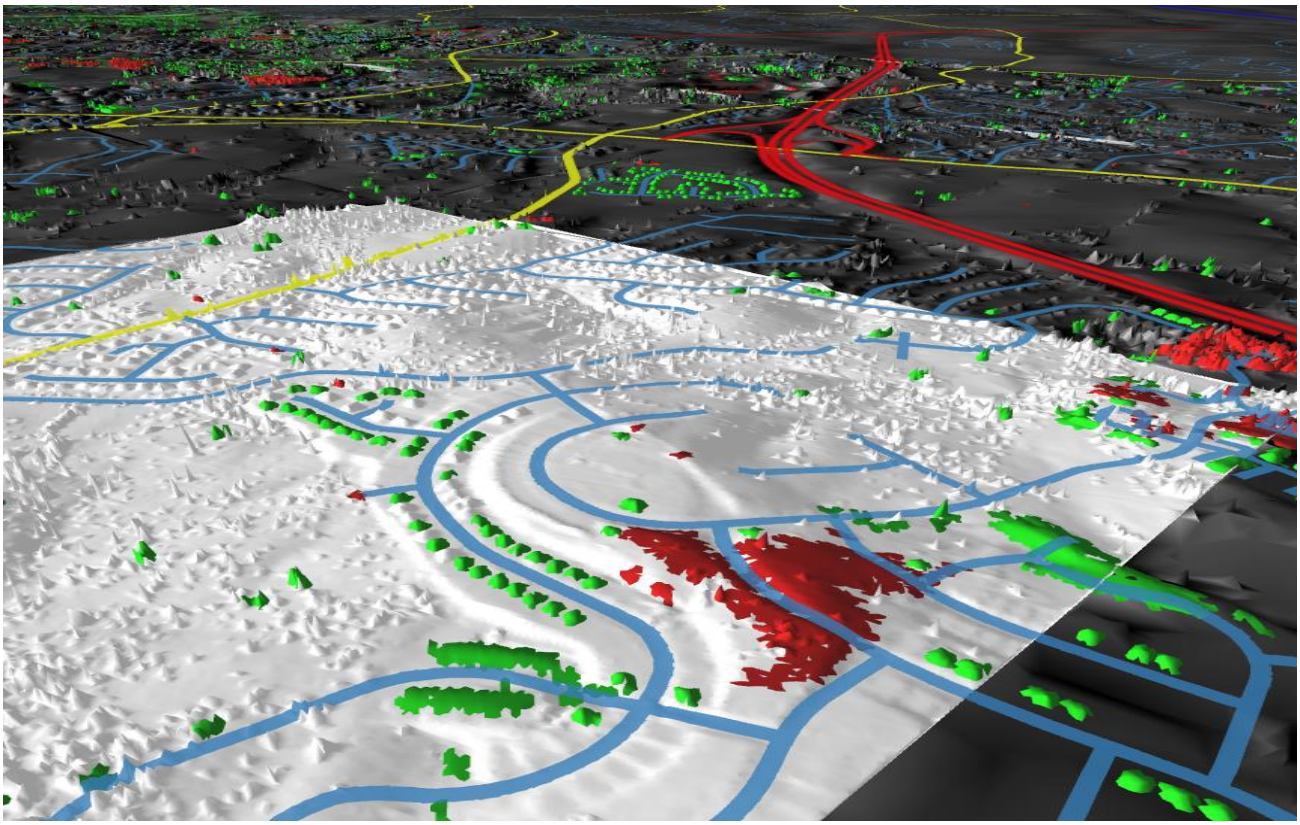

Fig. 4. An example of an interactive visualization environment for analysis of detected change models. The LIDAR scans being compared are separated by one year and are of a much lower resolution than would be collected the system described here. Visible here is a new residential housing development under construction. The small, evenly spaced green objects are new houses. The large green surfaces on either side and across the roads are where earth was built up to level the road. The large red surface in the center is where earth was excavated to level the land for further construction of the road and additional housing units.

Manual steering of the system from the interactive application can include directly influencing the goal maps for collection efforts, adjusting the sensitivity of the change detection decision making and aggregation ranges for change model creation/updating, and adjusting how quickly changes are allowed to propagate into the main battlefield model.

There is much potential for collaborative interaction in this application as well. Monitoring a large or active battlefield could easily require multiple analysts to meet the evaluation demand for incoming change events. Collaboration could be as simple as dividing the battlefield into sectors and assigning each to a separate analyst, or a more complicated division in which different types of changes are pushed to specialized analysts.

3.8 Collection Goal Mapping

The final component of the system is tasked with collection goal map creation and maintenance. The behavior of this component is based on two assumptions. First, that significant changes will realistically occur in multiple local regions as opposed to being evenly distributed across the entire environment. As such we create a goal map that associates priority values with regions on the battlefield model, areas with higher priorities receiving more frequent measurements from the airborne sensors. The second assumption is that the probability of a new change being detected in a region is

directly proportional to the number of changes that were recently detected nearby. As such, we calculate the local values for our goal map based on the distances from those regions to recent change events. By doing so we not only increase the chances of discovering new changes, but we also ensure that existing changes will be resampled as soon as possible, leading to more detailed change models, or more tracked locations for a moving object. There is a danger that this becomes a self-fulfilling cycle in that areas have more changes detected in them because they are sampled more often, and thus it is also important to ensure that regions of the battlefield do not grow stagnant from not being sampled often enough. This is easily rectified by increasing a modifier for the priority of a region steadily over time and resetting this modifier each time the region is sampled. See figure 4 for an example of what a collection goal map based on changes previously detected may look like when calculated based on distances to recent significant changes.

Much research has dealt with these particular issues, and many successful solutions have been developed. Matei et al.[14] conclude that "3D object recognition on LADAR data is matured to the point that it is ready for real large-scale systems."[14] Ousingsawat[15] presents a method for UAV path planning of areas with varied priorities and explores the use of multiple UAVs to meet their surveillance goals. Likewise, Ahmadzadeh[16] have developed a time-critical path planning algorithm aimed at enabling multiple UAVs to coordinate their surveillance activities using their (constrained) integrated motion sensing capabilities. An overview of the abilities, challenges, and effective strategies for control of multiple UAVs can be found in Zennaro[17].

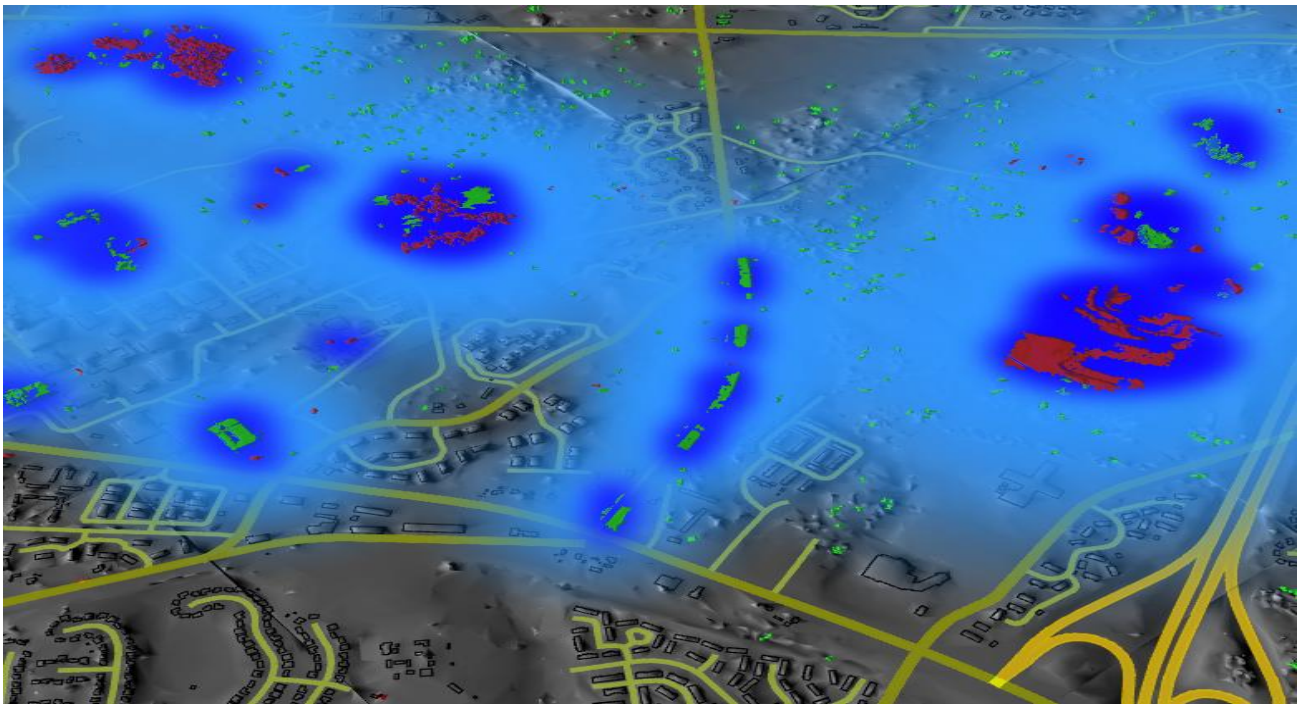

Fig. 5. A possible collection goal map based on previously detected changes (red and green). The darkest blue areas would be of highest priority for resampling by the airborne sensors, the lighter blue areas of lesser importance, and the unshaded areas would be of the lowest priority for resampling. By predicting that new changes will occur in the same general neighborhoods as previous changes, and guiding our sensors thusly, we can increase the probability of detecting new changes in a timely fashion.

4. DISCUSSION

The system described here could be applied to non-battlefield usage as well. A prime example of another application in which this type of time-critical change detection is crucial is border patrol. The United States Border Patrol already utilizes UAVs as "force multipliers," allowing deployment of fewer agents while maintaining their ability to detect and counter border intrusions.[18] The Intelligence Reform and Terrorism Prevention Act (P.L. 108-458) calls for

investigation into the further implementation of UAVs along the expansive U.S/Canada border, and requires The Department of Homeland Security to develop plans to monitor the U.S./Mexico border with UAVs. DHS has been testing UAVs for surveillance of the U.S./Mexico border since June 2004. Clearly there is great interest in using these platforms, due especially to their sensing abilities and loiter capability. (The Predator B used on the U.S./Mexico border can fly 15 times as long as a traditional helicopter before refueling)[18]

Other remote sensing technologies exist and are being developed that have the ability to enhance the reliability and increase the measurement and identification capabilities of this system beyond what solely LIDAR can provide. Synthetic Aperture Radar (SAR) is an existing sensor technology that has this potential. Among other benefits, SAR has the ability to penetrate through and sense beyond visual obstacles such as vegetation cover. This is attractive because it allows not only the surveillance of additional regions, but foils attempts to obstruct aerial observation through the use of smoke cover. Sandia National Labs has already developed *miniSAR*, a 30 pound SAR unit for use on small UAVs which boasts a 4 inch resolution.[19] With the ability to affordably deploy these powerful remote sensing technologies over the modern battlefield it will become increasingly important to develop systems that can assimilate measurements from multiple sources and technologies.

5. CONCLUSIONS

We have presented the framework for a system that combines several existing technologies and algorithms to reduce the massive amount of data generated from a fleet of airborne LIDAR sensors into a manageable stream of events which analysts can use to gain insight and report on battlefield changes as they develop. It is our hope that the design and ideas presented in this paper and will lead to further discussion on methods to fully utilize the powerful remote sensing and surveillance capabilities available in modern unmanned aerial vehicles. The integration of the techniques presented and referenced here and the further development of similar robust and powerful analytical tools have great potential to maximize the utility of available hardware and provide superior situational awareness both on the battlefield and elsewhere.

ACKNOWLEDGMENTS

This work is supported by the Army Research Office under contract no. W911NF-05-1-0232.

REFERENCES

[1] J.C. Fernandez, A. Singhania, J. Caceres, K.C. Slatton, M Starek, and R. Kumar, "An Overview of Lidar Point Cloud Processing Software," GEM Center Report No. Rep_2007-12-001, University of Florida, December 20, 2007.
[2] Quick Terrain Modeler, Applied Imagery / USA, http://www.appliedimagery.com/index.html.
[3] Mars Viewer 4.0, Merrick & Company / USA, http://www.merrick.com/servicelines/gis/mars.aspx.
[4] Polyworks, Innovmetric / Canada, http://www.innovmetric.com/Manufacturing/home.aspx
[5] Terrasolid Ltd., Finland, http://www.terrasolid.fi/.
[6] Fledermaus, Interactive Visualization Systems / Canada, http://www.ivs3d.com/.
[7] Matlab, MathWorks / United States, http://www.mathworks.com/.
[8] Calder, B., "Automatic Statistical Processing of Multibeam Echosounder Data," International Hydrographic Review, Vol. 4, 53-68, 2003.
[9] D. Mallace and G. Lindsey, Multibeam processing – the end to manual editing?, IVS3D Whitepaper, www.ivs3d.com.
[10] Suveg, Ildiko and Vosselman, George, "Reconstruction of 3D building models from aerial images and maps", ISPRS Journal of Photogrammetry and Remote SensingVolume 58, Issues 3-4, Integration of Geodata and Imagery for Automated Refinement and Update of Spatial Databases, January 2004, Pages 202-224.
[11] Michael Garland, Paul S. Heckbert, Surface Simplification Using Quadric Error Metrics, Proceedings of SIGGRAPH 1997, p209-216, August 1997.

[12] Alexandru N. Vasile and Richard Marino, “Pose-independent automatic target detection and recognition using 3D LADAR data” Proc. SPIE Int. Soc. Opt. Eng. 5426, 67, 2004.
[13] Funkhouser, Thomas; Min, Patrick; Kazhdan, Michael; Chen, Joyce; Halderman, Alex; Dobkin, David; Jacobs, David, “A Search Engine for 3D Models” ACM Transactions on Graphics, 22(1), pp. 83-105, January 2003.
[14] Matei, B. C., Tan, Y., Sawhney, H. S., and Kuma, R. “Rapid and scalable 3D object recognition using LIDAR data”, Automatic Target Recognition XVI. Edited by Sadjadi, Firooz A.. Proceedings of the SPIE, Volume 6234, pp. 623401, 2006.
[15] Ousingsawat, Jarurat. “UAV Path Planning for Maximum Coverage Surveillance of Area with Different Priorities”, The 20th Conference of Mechanical Engineering Network of Thailand. Nakhon Ratchasima, Thailand. 18-20 October 2006
[16] Ahmadzadeh, A.; Jadbabaie, A.; Kumar, V.; Pappas, G.J., "Multi-UAV Cooperative Surveillance with Spatio-Temporal Specifications," *Decision and Control, 2006 45th IEEE Conference on* , vol., no., pp.5293-5298, 13-15 Dec. 2006
[17] Ryan, A.; Zennaro, M.; Howell, A.; Sengupta, R.; Hedrick, J.K., "An overview of emerging results in cooperative UAV control," *Decision and Control, 2004. CDC. 43rd IEEE Conference on* , vol.1, no., pp. 602-607 Vol.1, 14-17 Dec. 2004
[18] Bolkcom, Christopher and Nuñez-Neto, Blas, “Homeland Security: Unmanned Aerial Vehicles and Border Surveillance”, Congressional Research Report, November 17th, 2007, Congressional Research Service, Library of Congress, http://www.ilw.com/immigdaily/news/2007,1101-crs.pdf
[19] Sandia National Laboratories press release “Sandia’s miniSAR offers great promise for reconnaissance and precision-guided weapons” February 18, 2004
[20] Butkiewicz, Thomas; Chang, Remco; Wartell, Zachary; Ribarsky, William. “Visual Analysis and Semantic Exploration of Error Aware Urban Change Detection”. In submission to Eurographics/IEEE-VGTC Symposium on Visualization 2008. Volume 27, Number 3.