LIDAR-based Virtual Environment Study for Disaster Response Scenarios

Giang Bui, Prasad Calyam, Brittany Morago, Ronny Bazan Antequera, Trung Nguyen, Ye Duan
University of Missouri-Columbia
Email: {*gdb338*, *bagth5*, *rcb553*, *tqnxbb*}@*mail.missouri.edu*; {*calyamp*, *duanye*}@*missouri.edu*

***Abstract*—In the event of natural or man-made disasters, many videos may be collected by civilians and surveillance cameras that can be extremely useful for first responders trying to ascertain the extent of the damage. However, watching and analyzing numerous videos on separate screens can be a cumbersome task. Registering a set of 2D videos with a 3D model can provide an intuitive venue for viewing multiple videos simultaneously. In such a setup, it is likely that the user will want to work with the dynamic 3D environment from a remote location, requiring that videos be transferred over a network to be registered with a 3D model. In this paper, we propose combining the fields of computer vision, cloud computing, and high-speed networking to create a system that takes in HD videos, streams the data to a server where a dynamic 3D model is constructed, and provides a virtual scene navigation program for viewing the videos in a 3D scene from a mobile device. We test transferring the data of interest over different types of networks and processing the videos on various server configurations to determine the capabilities of such a system and the necessary requirements for it to provide a high-quality user experience.**

I. Introduction

During natural or man-made disasters, videos from many perspectives are collected by security cameras and civilian observers. This abundance of data can be helpful for officials and emergency responders who need to quickly ascertain the state of affairs. When the normal infrastructure starts breaking down, it can be difficult for the authorities deciding how to respond to access, observe, and determine the extent of the damage. However, it is becoming increasingly common for civilians to record videos of unfolding events on cell phones, making crowd-source information available in addition to video feeds from mounted security cameras. Unfortunately, handling such large collections of videos and performing this type of surveillance on a traditional 2D grid display can be quite challenging. It is difficult for users of such systems to perceive the spatio-temporal relationships between different video streams, understand the geographical context of the situation, track and analyze events as they unfold, and predict possible outcomes. On the other hand, *3D virtualizations of scenes* can help a great deal in obtaining such information, especially in disaster scenarios [1]. Allowing a model of a 3D scene to be viewed from remote locations by sending it over available networks to officials can aid in response organization. Large scale models can be created using 3D LIDAR (Light Detection and Ranging) scans that use a laser to determine distance to surfaces and reconstruct a scene [2]. This type of LIDAR data has been shown to be useful for assessing the aftermath of natural disasters since highly accurate scans can be obtained very quickly [3]. However, LIDAR data by nature can be large and computationally expensive to process so adequate resources must be available to take advantage of this rich source of information. Leveraging the elasticity of cloud computing and the resources of high-speed networks provides new opportunities for using LIDAR data for life-saving applications.

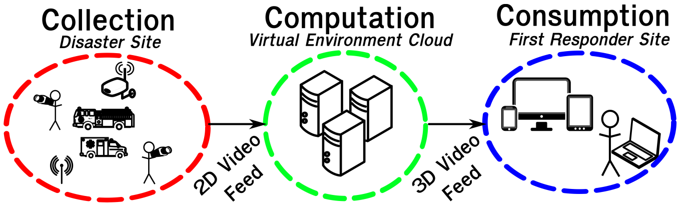

Fig. 1. Diagram showing overview of system. Videos are collected at disaster scene and transfered wirelessly to the server where their 3D poses are estimated. The virtual 3D environment is transferred from the server to thin clients for data consumption.

In this project, we study how these opportunities can be realized by working with a dynamic interactive 3D scene visualization system in which videos are captured at a disaster site, transferred to a server for on-demand processing and model construction, and viewed remotely with a thin-client. We look at provisioning options such as elastic compute to carry out these tasks. This overall process of *collection, computation, and consumption* is outlined in Figure 1. By seamlessly rendering dynamic video data from multiple cameras on top of a LIDAR point cloud, the system allows the users to view the recorded action in the context of a global 3D model from the viewpoints of any virtual camera. We test running our system over various application-driven overlay networks using concepts of software defined networking (SDN) [4], [5] to identify the network configuration requirements for processing and viewing 3D video and delivering high Quality of Service (QoS). We use a wireless overlay network (WON) to represent a standardly available network and a higher-speed network that simulates a fast overlay network made available on top of existing infrastructure in a disaster scenario that meets special needs, which we refer to as a disaster overlay network (DON). Videos must be able to be uploaded to the server over DON without encountering network congestion so that they can be processed on-demand in the cloud infrastructure and transferred via the same network to thin-client protocols for first responders to view.

The remainder of our paper is organized as follows. Section II reviews related work in the network and vision com-

This work was supported by the National Science Foundation under awards ACI-1245795 and CNS-1205658 and Cisco Systems. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or Cisco Systems.

munities. Section III summarizes our computer-vision based techniques for the computation stage referred to in Figure 1 in which videos are registed with LIDAR range scans to create 3D virtual representations of videos. Section IV details the technology we use in our virtual environment setup, our system requirements, and our data acquisition process as well as our experimental setup. In Section V we report the results of our study and Section VI concludes our work.

II. Literature Review

3D representations of a disaster scenario can be transferred over wireless networks to remote locations for better scene understanding than what a set of disjointed videos and photographs would provide. Research on how to set up mobile networks in a disaster scenario and on how to create 3D models and simulations using LIDAR data have provided strong foundations for accomplishing this, but to the best of our knowledge, they have only been studied as separate topics. It is important in disaster scenarios to be able to transfer and share data amongst emergency response crews, administrative officials, and medical professionals for planning and rescue purposes. Several groups have investigated ways in which wireless networks can be set up and utilized for communication in the event of an emergency [6]. Chissungo et. al [7] have studied using Wireless Mesh Networks to transfer medical information throughout disaster zones in situations where wired networks are damaged. Witkowski et. al [8] set up mobile ad hoc networks that allow humans to communicate with robots being used to explore the aftermath of a disaster. As these types of studies have become more popular, the speed of message delivery over wireless networks and the energy efficiency of these on-the-fly network setups has been prioritized and explored [9]. In this work, we study achieving similar goals of high-speed communication with DON.

The fact that fusing 2D imagery with 3D LIDAR scans can be used to create large scale, photorealistic 3D models very quickly and easily with a high degree of accuracy has motivated several research projects in the computer vision field. Many groups have focused on performing registration on urban data which has an abundance of regularized features such as line segments and arcs that can be matched across dimensions [10]. Mutual information can also be used for direct 2D-3D registration in which various properties of a scan such as laser reflectivity or point cloud height are visualized in 2D [11], [12]. If the scanner has a build in camera, keypoint features can be matched in 2D to obtain an camera pose estimate that can be refined with 3D normal or edge information [2], [13].

These methods make available visually rich information for scene understanding but are computationally intensive and can benefit performance-wise from high-speed networking and cloud computing resources. By transferring collected 2D and 3D data to a remote server for real-time computation and processing and transferring the final fused results to mobile devices for consumption, a whole new range of use cases for LIDAR data in disaster scenarios becomes possible.

III. 2D-3D Registration

In order to register a video with the LIDAR range scan, we must calculate the camera poses for video frames in relation to the 3D point cloud. This entails matching a video frame to LIDAR photographs whose 3D correspondences are known and solving for the cameras projection matrix. To initially determine the 2D-3D relationship between the LIDAR photographs and the LIDAR scan, we have a pre-processing stage during which we map 2D pixels to 3D points. The mapping between the camera and the scan is known from a provided file giving the camera's focal length in pixels and rotation and translation. Using this information, each 3D point is projected onto each image plane to find its corresponding 2D point. The entire point cloud is projected onto each image onto each image once and the 2D-3D correspondences are saved on the server.

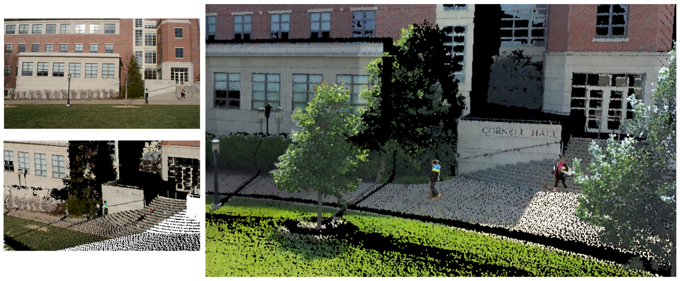

Fig. 2. Creating 3D planes for dynamic objects. *Top Left:* 2D video frame. *Bottom Left:* Video frame projected onto range scan without using our method for modeling moving objects. *Right:* 3D planes constructed for moving objects identified in video using our modeling method.

Once we have this information, we perform SIFT (Scale Invariant Feature Transform) matching between video frames and LIDAR photographs [14] and obtain a set of 2D-3D keypoint matches between the video frame and the LIDAR scan. We use this set of 2D-3D correspondences to calculate the projection matrix, P, of the camera using the six-point algorithm [15]. The projection matrix maps 3D LIDAR scan points to 2D video frame pixels, fusing the two modalities.

When a moving object, such as a person walking, that was not scanned is present in a video it will be projected onto an incorrect location in the 3D space because there is no structure that corresponds to it. These errors are very apparent when the user starts changing perspectives away from the original camera's viewpoint, as is demonstrated in Figure 2, Bottom Left.

To handle such situations, we segment out the motion in videos using the Mixture of Gaussians (MOG) algorithm [16] and add 3D planes to the virtual environment to "catch" the projection of these new entities. MOG yields a binary image with the motion segmented from the background. The connected components algorithm is applied to the MOG image to create cohesive segments. We scan this image starting from the bottom row of pixels to find the lowest point in each moving segment and identify its matching 3D point. Assuming that the moving object is touching the ground, this 3D point is the correct location for the bottom of the segmented object. New 3D points with the same depth as the bottom point and varying heights are created and projected onto the MOG image. If they fall within the segmented portion of the image, they correspond to a moving object that was not scanned and are added into the 3D space with the corresponding color information from the original video frame. The result of performing these steps is shown in Figure 2, Right.

IV. Experimental Methodology

A. Experimental Setup

Our testbed setup consists of clients connected to WON (∼10Mbps) that represent a standardly available campus enterprise network for QoS priorities and a compute manager VMware Horizon View© connected over a higher-speed campus research network DON (∼600Mbps). We emulate a network made available in a disaster scenario in which these two networks can be used in parallel. By pooling resources, our collection, computation, and consumption steps can be employed effectively by first responders. We have a virtual server setup with 6vCPU (12GHz) and 16 GB of memory with Windows 2008R2 64bits installed. Our physical server has 2 processors Intel Xeon Processor E5-2640 v2, 8 cores each for a total of 16 cores. The clients have a Windows 7 Enterprise 64 bits O.S. installed. Our clients are able to stream data to the server by using `curl` Linux utility functionality that is authenticated by the FTP server in the virtual server.

To obtain a 3D model for our location of interest, we use a Leica C10 HDS LIDAR scanner that provides a high-resolution point cloud of a scene and 2D images using a built-in camera. We also capture multiple video streams of people walking around the university campus with HD video cameras. This video data needs to be transferred in real-time to the HPC server for data processing.

B. Design of Experiments

We separately evaluate the performance for the three stages of our system shown in Figure 1, i.e. collection and transferring the 3D scan and video files, computing the 2D-3D data fusion, and consumption by the user to receive 3D scenes and multiple videos for virtual navigation and video analysis. The goal is to obtain real-time (or near real-time) responses for all of these tasks. We also experiment with scaling up the amount of data transferred to see how many videos we can handle and how large the 3D model data can be, depending on the hardware used.

To simulate the collecting and transferring of any number of real-time video streams, we first obtain several HD videos on campus. These videos are stored on a laptop and a varying number of duplicates are sent over the network simultaneously to tax the system. Our goal is to observe what happens to the system when one verses many videos become available and need to be viewed. Individual video frames are transferred sequentially to mimic real-time video capture. The 2D-3D registration and video motion analysis stages are performed on the server.

For the final consumption stage, the large 3D model only needs to be transferred over the network to the remote device one time when it is first requested. If the user wishes to a view a different location, a new model will need to be sent to the mobile device (laptop, tablet, cellular phone, etc.).

V. Study Results

We studied the collection, computation, and consumption sections of our pipeline individually. All of our tests were performed three times, and in this section we report the averages of these tests as our final results.

A. Collection

We tested sending varying file sizes over the server (52, 105, 210, and 316 MB) to account for situations where different definition videos are streamed from client connected to WON and DON. The transfer times in seconds for these tests are shown in Figure 3. The maximum number of videos we tested sending at once is five because our server has six cores and cannot process more than that number of videos at once. Despite the fact that these are relatively small-scale tests, we still get a good sense from the charts how communication time will increase as the number of videos rises. These tests also show that DON is able to transfer data about 10 times faster than WON and would be very beneficial in a disaster scenario where timely information sharing is key.

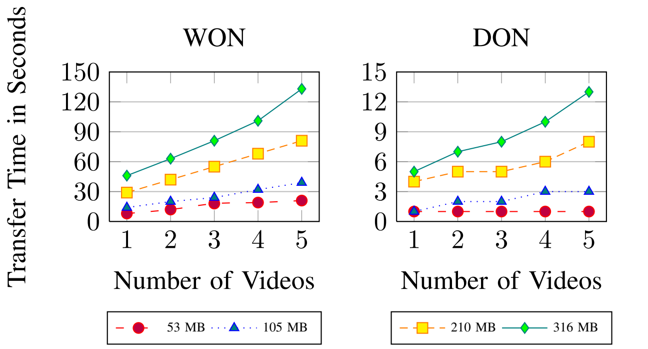

Fig. 3. Collection stage transfer times for varying video sizes. *Left:* Transfer times for WON. *Right:* Transfer times for DON.

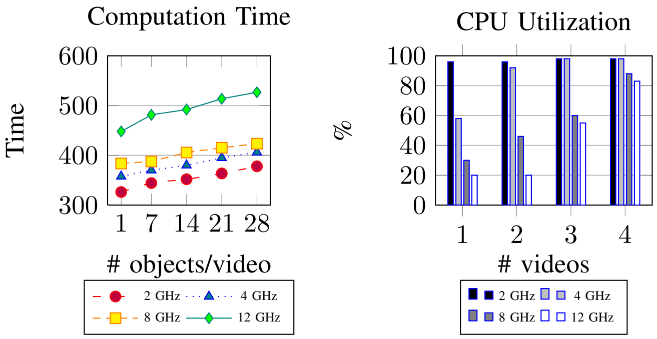

Fig. 4. Performance during the computation stage. *Left:* Time measured in seconds required to compute 3D pose for increasing number of dynamic objects in videos on different server configurations. *Right:* CPU utilization on server during computation stage for different configurations.

B. Computation

We modified the vCPU capacity of our virtual server with 2, 4, 8 and 12 GHz, testing the processing times in seconds for videos containing 1-28 moving objects. Each dynamic object in every video needs to be identified, segmented from the static background, and modeled in 3D so we are interested in what happens to our overall performance as more objects are recorded. We stopped at 28 objects because this seems to be a reasonable limit on the maximum number of people that will be captured in a typical camera's field of view and be able to be separately identified and modeled as individual objects in 3D. We also tested the system's performance when processing 1 to 4 videos of 185MB each with the same content simultaneously and looked at the CPU percentage utilization. These results are all shown in Figure 4. We observe here that the system becomes saturated when processing four videos and can see what will happen as more videos are added to

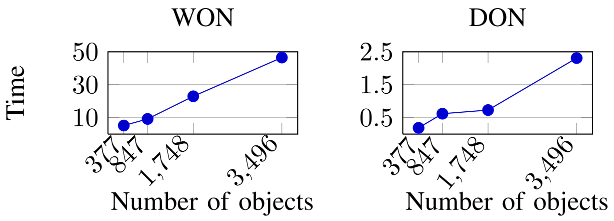

Fig. 5. Consumption evaluation measured in seconds. Transfer times for dynamic 3D objects captured in videos to be sent to remote device for viewing. *Left:* Transfer times for WON. *Right:* Transfer times for DON.

the system. We gain the greatest boost in performance when increasing from two to four videos.

C. Consumption

During the consumption stage, the clients need to download the files containing 3D information from the server. In the case that a client is connected to the server via WON, this process is time consuming compared with a virtual desktop accessed from a thin-client (hardware/software). For both cases, Teradici PCoIP protocol© is used for remote access. A comparison of file transfer times in seconds between a physical client connected to WON and using a virtual desktop setup on a server connected to DON is shown in Figure 5. We tested transferring 3D data files for between 377 and 3,496 individual moving objects simultaneously to significantly stress the system and to find out how much information can be processed in a timely manner if the disaster site is very congested with people and cameras. We can see that using DON, thousands of moving objects can be transferred and displayed in a matter of seconds, making this setup great for first responders needing to rapidly sift through vast information from the disaster scene.

Our final stage of testing looks at the actual user experience during the consumption stage. We evaluated the data transfer times in Kbps for running various programs over the thin-client. We compared the performance of our 3D video program to everyday programs that most people are familiar with such as Excel and Internet Explorer on four different types and speeds of networks, as shown in Figure 6. We can see that the 3D video program requires a tremendous amount of resources for processing because it contains vast amounts of rich visual information even after encoding. Ideally, emergency responders will be able to use thin-clients for 3D video analysis to avoid potentially long download times, 3D viewing software setup and speed up data acquisition.

VI. Conclusion

In this paper, we combined the fields of computer vision, cloud computing, and high-speed networking to create 3D visualizations of disaster scenarios for scene understanding. We presented results for the three stages of a system that collects videos of a scene, performs the necessary computations to register them with a 3D LIDAR scan on a remote server, and transfers them to mobile devices for consumption and scene viewing. Our tests demonstrate how a high-speed network used in a disaster scenario can greatly increase the speed at which data can be shared amongst officials and emergency responders spread out over numerous locations making crucial decisions. We also show that multiple videos with recordings of many moving objects such as people and cars can be sent to a cloud server, registered, and viewed in 3D simultaneously using thin-clients by emergency response teams. Thus, we avoid having to download all the data and necessary analysis software, and increase the usefulness of this system in critical moments when officials need to make quick, informed decisions to save lives.

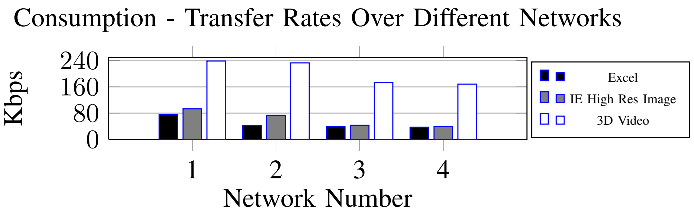

Fig. 6. Comparing user experience of our program to other standard programs over the thin-client during the consumption stage. Network 1 the Campus Wireless with a bandwidth of 8-9 Mbps/10 Mbps. Network 2 is a Wired Lab with 7 Mbps/7 Mbps. Network 3 is a Wired Lab with 5 Mbps/3 Mbps, and Network 4 is a Wired Lab with 3 Mbps/1 Mbps.

References

[1] N. Schurr, J. Marecki, M. Tambe, P. Scerri, N. Kasinadhuni, and J. P. Lewis, "The future of disaster response: Humans working with multiagent teams using defacto." in *AAAI Spring Symposium*, 2005.

[2] B. Morago, G. Bui, and Y. Duan, "Integrating lidar range scans and photographs with temporal changes," in *CVPRW*, 2014.

[3] M. Kwan and D. M. Ransberger, "Lidar assisted emergency response: Detection of transport network obstructions caused by major disasters," *Computers, Environment and Urban Systems*, vol. 34, no. 3, 2010.

[4] S. Seetharam, P. Calyam, and T. Beyene, "ADON: Application-Driven Overla Network-as-a-Service for Data-Intensive Science," Master's thesis, University of Missouri-Columbia, 2014.

[5] P. Calyam, A. Berryman, E. Saule, H. Subramoni, P. Schopis, G. Springer, U. Catalyurek, and D. K. Panda, "Wide-area Overlay Networking to Manage Science DMZ Accelerated Flows." IEEE, 2014.

[6] S. Kumar, R. K. Rathy, and D. Pandey, "Design of an ad-hoc network model for disaster recovery scenario using various routing protocols," in *Advances in Computing, Communication and Control*. ACM, 2009.

[7] E. Chissungo, H. Le, and E. Blake, "An electronic health application for disaster recovery," in *Information and Communication Technology*. ACM, 2010.

[8] U. Witkowski, E. Habbal, M. A. Mostafa, S. Herbrechtsmeier, A. Tanoto, J. Penders, L. Alboul, and V. Gazi, "Ad-hoc network communication infrastructure for multi-robot systems in disaster scenarios," in *IARP/EURON Workshop on Robotics for Risky Interventions and Environmental Surveillance*, 2008.

[9] N. K. Ray and A. K. Turuk, "A framework for disaster management using wireless ad hoc networks," in *Communication, Computing & Security*. ACM, 2011.

[10] I. Stamos, L. Liu, C. Chen, G. Wolberg, G. Yu, and S. Zokai, "Integrating automated range registration with multiview geometry for the photorealistic modeling of large-scale scenes," *IJCV*, 2008.

[11] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice, "Automatic targetless extrinsic calibration of a 3d lidar and camera by maximizing mutual information," *Proc. AAAI National Conference on Artifical Intelligence*, 2012.

[12] A. Mastin, J. Kepner, and J. Fisher, "Automatic registration of LIDAR and optical images of urban scenes," in *CVPR*. IEEE, 2009.

[13] G. Yang, J. Becker, and C. Stewart, "Estimating the location of a camera with respect to a 3d model," in *3DIM*. IEEE, 2007.

[14] D. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, 2004.

[15] R. Hartley and A. Zisserman, *Multiple View Geometry*. Cambridge, United Kingdom: Cambridge University Press, 2010.

[16] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," in *CVPR*. IEEE, 1999.