Mosaic-based Modeling and Rendering of Large-Scale Dynamic Scenes for Internet Applications

Edgardo Molina[a], Hao Tang[a], Zhigang Zhu[a], Olga Mendoza[b]
[a]Department of Computer Science, The City College of New York, New York, NY 10031
[b]Air Force Research Laboratory, WPAFB, Ohio 45433-7318, USA

Abstract

3D models of large-scale scenes available on the Internet today are largely manually created. Thus it takes a long time to create them for cities and update them as those cities that are already modeled continue to change. Multiple parallel-perspective mosaics can be generated from video automatically and more efficiently and can be used to reconstruct 3D scenes faster. A lot of video currently exists on the Internet, many of which are of aerial scenes that are currently not being utilized to their full potential. Reconstructing scenes from video provides the benefit of containing dynamic activity and texture information of the scenes in addition to the 3D structure data.

1. Introduction

Today we can find many professional stock video websites selling aerial footage taken with standard and high definition cameras shot from helicopters and light aircraft. On websites like YouTube.com we can find many amateur and enthusiast user-generated aerial videos which are taken using consumer-grade video cameras and web cameras mounted on devices ranging from helium balloons to RC planes and RC helicopters.

So far the Internet has made sharing and tagging this video content possible and searchable textually. But very little or nothing has been done to truly understand and classify a videos activity content, its camera motion, and how its scenery and dynamic content should be presented. Understanding a videos camera motion is crucial for scene understanding and reconstruction. Scene understanding allows us to represent a video's content in alternative forms. One way to represent these videos is to use a layered approach in which detected activities and detected landmarks are extracted onto separate layers from their background layer. A richer representation, in particular for aerial videos, is to represent the videos in 3D by applying 3D reconstruction methods.

The availability of large-scale scene 3D models on the Internet is currently limited compared to the recent rise in video availability. Applications such as Microsoft Virtual Earth, Google Earth and the 3D Warehouse community are making more 3D models available to users on the Internet, but these models are largely manually created. It is possible to obtain real world 3D data and models through the use of LIDAR or Laser Range sensors. But this is currently an expensive option that is only cost effective, or within budget, for a limited number of users and institutions. Also, models that are manually created or captured using specialized sensors will become stale over time due to the expense and time involved in updating the models; For example, a model of any city needs to constantly change as building and structures are torn down, expanded and built. Video, on the other hand, is inexpensive, widely available and can provide 3D models of large-scale scenes faster (and potentially in real-time) than specialized sensors or than they can be manually modeled. In addition, aerial videos and ground videos on moving platforms can be used directly for 3D scene reconstruction.

In this paper we present our mosaic-based approach for reconstructing large-scale dynamic scenes and discuss the potential it has for interactive 3D applications on the Internet. In Section 2 we discuss related work on large-scale scene reconstruction. Section 3 introduces the geometry for multi-view dynamic pushbroom mosaics and the two-phase system we developed for large-scale scene reconstruction. Section 3.1 describes Phase I: how an input video is converted to multi-view dynamic pushbroom mosaics; Section 3.2 describes Phase II: how multi-view pushbroom mosaics are used for content extraction and 3D reconstruction. Section 4 elaborates on a few compelling scenarios where fast reconstruction and rendering of large scale dynamic scenes can be very beneficial. In Section 5 we present how the large-scale scenes can be visualized, using multi-view pushbroom mosaics in Section 5.1, and using the CB3M representation in Section 5.2. And in Section 6 we discuss our ongoing work and future considerations.

2. Related Work

Google Earth [7] and Microsoft Virtual Earth [12] map the earth by superimposing images obtained from satellite imagery, aerial photography and integrating Geographic

Information System's data into their respective applications. Many buildings and structures from around the world now have detailed 3D structures, however, these 3D models are created manually and the number of city models available is limited.

Two intuitive methods to obtain a 3D urban model are to use a 3D sensor or to reconstruct a 3D model using multiple overlapping aerial images. 3D sensors such as LIDAR can provide 3D information in point cloud format. In order to refine 3D data models from LIDAR data, V. Verma, et. al. [18] propose a method that detects and refines 3D urban models by automatically recognizing building roofs. Most of the recent work in building detection and reconstruction has focused on the stereo or multi-view analysis [2, 4]. Z. Kim and R. Nevatia [10] have developed a framework to detect and model complex buildings by a hypothesis and verification procedure from multiple images.

Combining multiple and different data resources to generate 3D urban models is an active field of research. W. Zhao, et. al. [20] propose a general framework for aligning continuous video onto 3D sensor data that can be used to reconstruct 3D urban models with texture maps. A. Zakhor, et. al. [21] propose a system that incorporates two different data sources. It uses LIDAR to obtain an urban model, and uses both aerial images and a ground laser scanner to obtain texture of buildings and façades, and integrates the two models to obtain a more accurate urban model. I. Stamos, et. al. [16] propose a framework that integrates automated range registration with multi-view geometry for photorealistic modeling of large-scale urban scenes.

Research has also been proposed to supply realistically textured 3D city models at ground level. M. Pollefeys, et. al. [14] present a system for automatic, geo-registered, real-time (using GPU) 3D reconstruction from video of ground urban scenes. N. Cornelis, et. al. [6] have developed a complete system for turning forward-looking stereo video captured by a moving car into a model from which a virtual drive-through a city street can be rendered, which is useful for pre-visualizing upcoming traffic situations in car navigation systems.

Sarnoff corporation has created the TerraSight system [17] that makes it possible to overlay mosaics and videos, both recorded and live, over geographic models.

A. Rav-Acha, et. al. [15] have developed a method to compute depth and create mosaics and renderings of long video sequences, using the MAD and X-Slits approach they have developed.

3. Multi-View Pushbroom Mosaicing

We first give an introduction to the geometry of pushbroom stereo [22] mosaics for a static scene. If we assume the motion of a camera is a 1D translation and the optical axis is perpendicular to the motion, then we can generate two spatio-temporal images (mosaics) by extracting two scanlines of pixels from each frame. Figure 1 illustrates the geometry. A 3D point P on a target is first seen through the leading edge of an image frame when the camera is at location L_1. If the point P is static, we can expect to see it through the trailing edge of an image frame when the camera is at location L_2. Therefore, the distance in the y direction (baseline B_y) between two locations L_1 and L_2 is proportional to the depth of P. That is, if object is close to camera, baseline is small. Otherwise, it's large. Therefore, the pushbroom geometry has an adaptive baseline (different from perspective geometry) and the depth resolution is uniform (better than perspective geometry). By searching for the correspondence of point P on the two mosaics, the scaled baseline (the displacement of two corresponding points in the y direction on the mosaics) can be measured and the depth of point P can be computed.

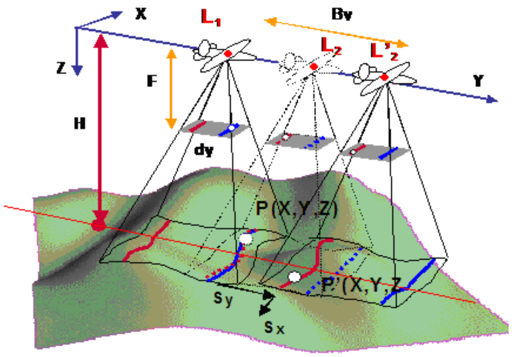

Figure 1: Dynamic Pushbroom Mosaic Geometry. Two mosaics are created from the leading and trailing edges of image frames. A static point P will be seen at camera positions L_1 and L_2. A moving point will be seen in camera positions L_1 and L'_2.

However, if point P moves during that time, the camera needs to be at a different location L'_2 to see this moving point (P') through its trailing edge. That is, the measured scaled baseline from stereo correspondence is larger in this case (for Figure 1) and reconstructed depth is larger than the true depth. Therefore, compared with the neighborhood area (static background area) of point P, the depth of P is a 3D anomaly and can be used to identify P as a moving point.

Figure 2 shows the two-phase system for generating multiple pushbroom mosaics and reconstructing 3D scenes and is described in the following sections.

3.1. Phase I: pushbroom mosaicing

The input to our system is a video sequence taken from a camera in motion with a dominant direction and its optical axis perpendicular to the motion. The outputs of

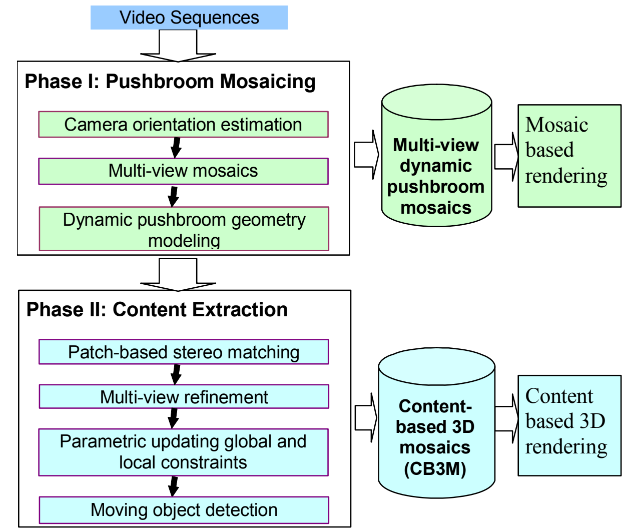
Video Sequences
Phase I: Pushbroom Mosaicing
Camera orientation estimation
Multi-view mosaics
Dynamic pushbroom geometry modeling
Multi-view dynamic pushbroom mosaics
Mosaic based rendering
Phase II: Content Extraction
Patch-based stereo matching
Multi-view refinement
Parametric updating global and local constraints
Moving object detection
Content-based 3D mosaics (CB3M)
Content based 3D rendering

Figure 2: Diagram of the two-phase system for creating pushbroom mosaics and 3D reconstruction.

Phase I are a set of multi-view pushbroom mosaics, which can be rendered and visualized directly and used in phase II for content extraction and 3D reconstruction.

3.1.1 Camera orientation estimation

The first step is to perform camera orientation estimation on the video sequence. The inter-frame motion parameters can be calculated using bundle adjustments (up to a scale) or obtained with external orientation measurements. In this paper, we provide examples of parallel-perspective mosaics with parallel projections in a dominant motion direction. But the principles can be applied to other types of ego-motion, such as circular motion or more general motion.

3.1.2 Generation of pushbroom mosaics

Once the camera movement is analyzed we can determine the dominant direction of the camera's motion. In the ideal case, pushbroom mosaics are constructed by accumulating the scanlines of the frames that are perpendicular to the direction of motion. As an example, by taking the center scanline of all frames we construct a single mosaic with a nadir view of a scene in parallel-perspective projection. That is parallel in the direction of motion and perspective along the scanline. Figure 3 shows how we can build multiple mosaics by using multiple scanlines on all frames. The multi-view pushbroom mosaics constructed in this way will be in stereo correspondence with epipolar lines along the direction of the camera motion. Moving targets are also preserved in a pair of mosaics as spatio-temporal features. Typically, they violate the epipolar geometry and exhibit large visual motion between the two mosaics due to motion accumulation [8].

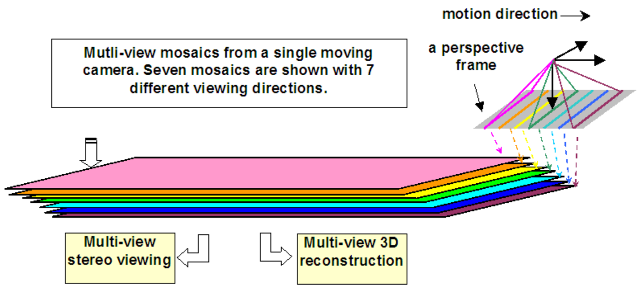
motion direction
Mutli-view mosaics from a single moving camera. Seven mosaics are shown with 7 different viewing directions.
a perspective frame
Multi-view stereo viewing
Multi-view 3D reconstruction

Figure 3: Multiple parallel-perspective pushbroom mosaics.

A pair of stereo mosaics is an efficient representation for both 3D structures and target movements. However, stereo matching will be difficult due to the largely separated parallel views of the stereo pair. Therefore, multi-view mosaics (more than 2) are generated, each of them with a set of parallel rays whose viewing direction is between the leading and the trailing edges (Figure 3). This method provides at least two benefits. First, it eases the stereo correspondence problem in the same way as multi-baseline stereo [13], particularly for improving accuracy of 3D estimation and handling occlusion. Second, multiple mosaics also increase the possibility to detect moving targets with unusual movements and also to distinguish the movements of the specified targets (e.g., ground vehicles) from those of trees or flags in wind. In the next Section (Phase II), we will discuss a method to extract both of the 3D buildings and moving targets from the stereo mosaics.

Real-time mosaicing It is possible to construct the multiple pushbroom mosaics in real-time using real-time orientation estimation algorithms. Therefore, a live feed of video can be processed as frames are received, and the multiple mosaics can also be generated as each new frame contributes a scanline of data. The mosaic images can dynamically grow as more spatial area is covered by the video feed.

3.2. Phase II: mosaic-based 3D reconstruction

To carry out stereo matching, we apply a segmentation-based stereo match algorithm, with two geometrical constraints. The basic workflow is given in phase II of Figure 2.

3.2.1 Patch-based stereo matching

First, we perform color segmentation [5] on the reference mosaic and each homogenous color patch is approximated to be a planar surface.

Interest points [8] are defined as those with large curvatures and are extracted along the boundary of each homogenous patch. For each pair of mosaics (reference mosaic and any other mosaic), a window-based stereo match method is performed on each interest point on each homogenous patch and a plane structure is fitted for each patch using RANSAC. In performing correlation, only those points on the patch are used to avoid the mixture depth problem.

3.2.2 Multi-view refinement

Suppose N+1 pushbroom mosaics (i.e. N pairs of consecutive mosaics) are constructed in the previous Section 3.1. For each homogenous patch, one set of plane parameters is computed based on one pair of mosaics. Therefore, N sets of plane parameters can be obtained from the N pairs mosaics. Only one of N plane estimates is selected to be an initial result with the minimal match cost function (Sum Squared Difference value) that is defined to be color similarity between target mosaic and warped reference mosaic according to the plane's parameters. An initial plane structure set is constructed by inserting initial plane structures of all patches.

3.2.3 Parametric updating global and local constraints

In order to further refine the initial 3D model, we explore two geometric constraints: local and global scene constraints. Figures 4 and Figure 6 show the resulting depth maps for two scenes.

Global scene constraint In many city scenes, there are one or more dominant directions of planar surfaces. For example, there exist three dominant planes (mutual perpendicular) in cities (e.g., New York City) and a dominant ground surface plane in suburban scenes.

An agglomerative clustering method is performed in the initial plane structure set to automatically vote for the dominant plane directions. These dominant directions are used as good candidates of estimation of plane surfaces so that some global scene constraint can be used to fix some unreliable matches.

Local scene constraint A neighboring-plane parameter hypothesis approach is carried out for each patch. That is, for each patch, plane structures of its neighborhood patch (including the patch itself) are used as hypotheses, and matching costs are used as measurement criteria in the testing step. Therefore, a plane structure with the best matching cost is selected to be the final plane parameters.

3.2.4 Moving object detection

Moving objects in the aerial image can be divided into two categories. In the first case, moving object patches move along epipolar lines, but they appear to be "floating" in air or below the surrounding ground, with depth discontinuities all around it. In other words, they can be identified by checking for 3D anomalies.

In the second (general) type of cases, most of the moving targets are not exactly on the direction of the camera's motion and a good matching cost can not be found for those patches. Therefore, for each of these patches, we can always perform a 2D-range search within its neighborhood area. If a good match (i.e., with a small Sum Squared Difference value) is found within the 2D search range, then the region is marked as a *moving* object. In Figure 4 (d), the areas that are numbered are detected moving objects.

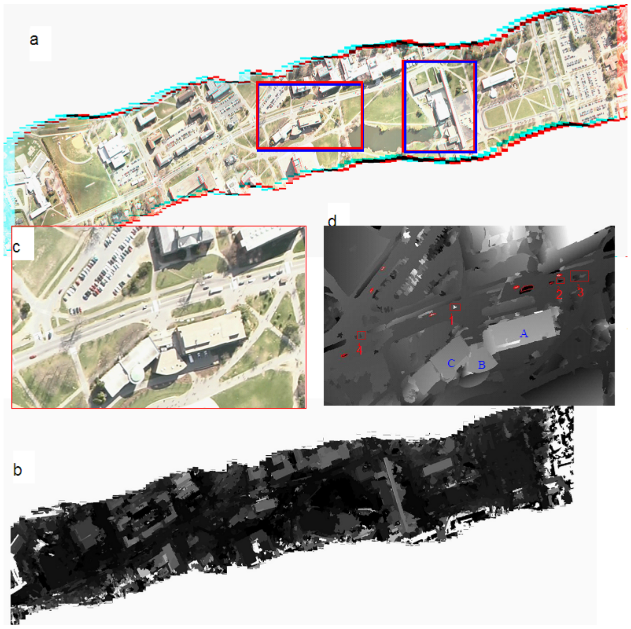

Figure 4: (a) Stereo mosaics for a campus scene. (b) depth map (c) section of the mosaic (d) depth map of c with numbered moving objects, and lettered plane patches.

3.2.5 Content-based 3D mosaic representation

A content-based 3D mosaic (CB3M) representation is a set of video object (VO) primitives (patches) that are defined as

$$\mathbf{CB3M} = \{VO_i, i=1, \ldots, N\} = \{ (c_i, \mathbf{b_i}, \mathbf{n_i}, \mathbf{m_i}), i=1, \ldots, N\}$$

where (1) N is the number of VOs, i.e., "homogeneous" color patches (regions) before region merging; (2) c_i is the color (3 bytes) of the *ith* region; (3) $\mathbf{b_i}$ is the 2D boundary of the *ith* region in the left mosaic, chain-coded as $\mathbf{b_i} = \{(x_0, y_0), K_i, b1, b2, \ldots b_{K_i}\}$, where the starting point (x_0,y_0) has 4 bytes, and each chain code has 3 bits. K_i is the number of boundary points (which needs 4 bytes each) and $K = \Sigma K_i$ is the total for all regions; (4) $\mathbf{n_i} = (n_x, n_y, n_z, d)$ represents the plane parameters of the region in 3D, 4 bytes for each parameter; and (5) $\mathbf{m_i}$ represents the L motion parameters of the region if in motion (e.g. L =2 for 2D translation on the ground).

Note that the VO primitives are those patches before region merging in order to preserve the color information. However, the plane parameters are obtained after multiple

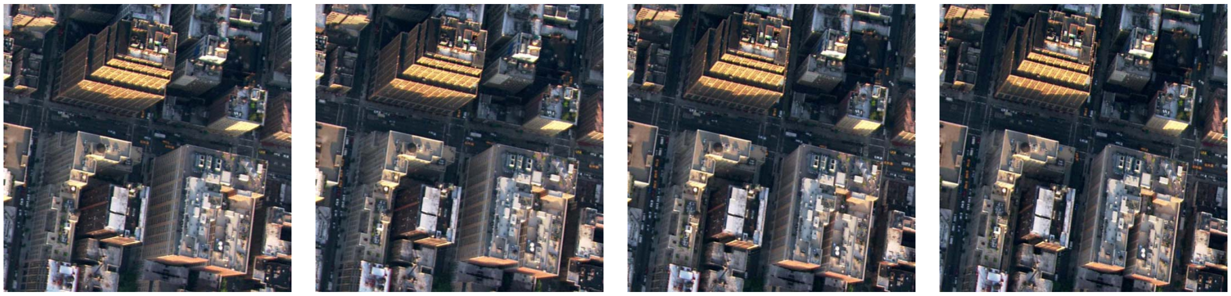

Figure 5: Four parallel-perspective views with different viewing angles over a close-up of the New York City scene. Parallax is preserved as well as the dynamic motion of moving vehicles.

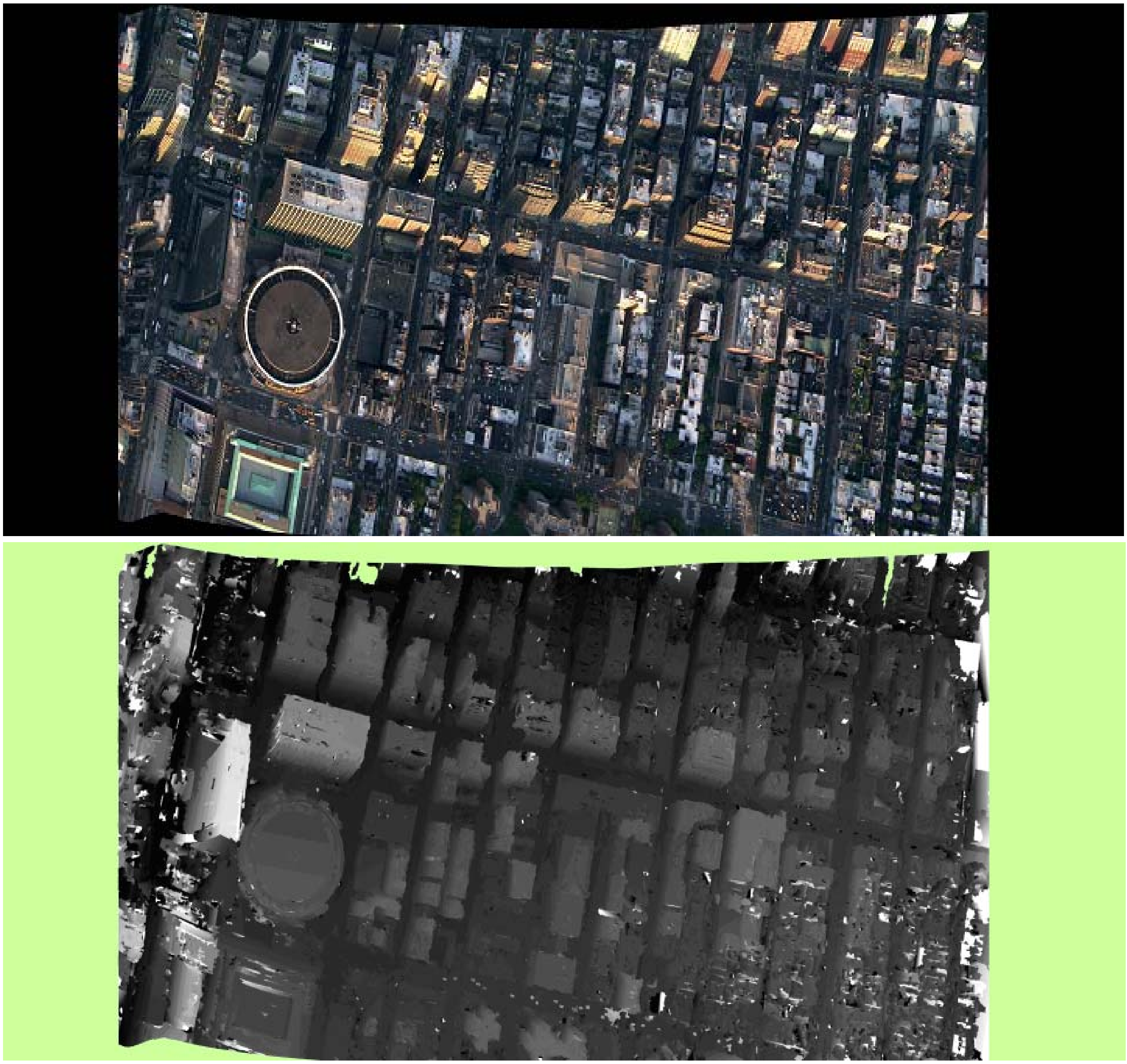

Figure 6: Mosaic of New York City scene (top) and corresponding depth map (bottom).

regions with different colors but on a same plane *surface* are merged.

The proposed CB3M representations are highly compressed visual representations for very long video sequences of dynamic 3D scenes. The representations could fit into the MPEG-4 standard [11], in which a scene is described as a composition of several Video Objects (VOs), encoded separately.

4. Visualizing Scenes: Application Scenarios

The ability to reconstruct and visualize large-scale scenes efficiently is crucial for the development of future interactive applications, particularly for applications on the Internet. These applications range from entertainment to emergency/disaster response.

4.1. Entertainment and mapping

One application for reconstructing entire cities and suburbs is purely for entertainment. Users will be able to better engage in virtual tourism. Users may want to do this if they are looking for a home to buy or a city to visit.

Internet mapping applications have become very popular for finding directions. Currently these applications are primarily 2D image based, with a few manually created 3D models. The next step is to provide real complete 3D models for these mapping applications. This would allow users to get an actual intuitive view of the locations they are finding and improve the directions that mapping services provide if the user is lost, especially if these are delivered to mobile devices.

4.2. City archival and planning

Historians and informational websites also have much to gain. Reconstructing large-scale scenes can be a frequent recurring task for the purposes of archiving how cities and areas continuously change over time. Continuous reconstruction also makes the data available fresher for all applications that use the 3D data. In order for continuous reconstruction to be viable, for non-security related applications, it must be inexpensive to do.

City planners and developers stand to benefit from large-scale scene reconstruction as they will have real 3D up-to-date data before starting projects.

4.3. Transportation and disaster response

Transportation systems can benefit by integrating their stationary cameras and other sensors with the 3D models. A user can get an intuitive view of traffic conditions as they plan out their commutes and transportation departments will be able to monitor congestion and plan in an intuitive way.

In cases of emergency or disaster response, the ability to quickly generate an updated 3D model of a large area will help response officials asses and respond to situations much faster than they would by visiting all areas (some of which may be unreachable via normal routes), or by viewing hundreds of hours of video. Citizens also gain access to view areas where they might have interests but no physical access.

During emergencies, disasters or congested traffic conditions a lot of aerial video is produced by news helicopters and government responders. Currently all of this video is broadcasted live over television, made available publicly on news websites, and as bandwidth increases it will be streamed live as well. This makes it possible for our mosaic-based reconstruction approach to be applied in real-time or near real-time to video streams.

4.4. Other applications

More generally, videos of large-scale scenes (for any purpose) can be easily captured by professionals or amateurs and delivered as is to end users, but they require a lot of storage space and bandwidth to store and deliver. Videos are also interacted with temporally, more commonly watched from start to end, requiring the length of runtime of the video for a user to obtain its information. A user can also skip to a particular point in time in the video, but this requires the user to know which point in the sequence is of interest and to have enough bandwidth to easily do this over the Internet (since video is often streamed, or preloaded). But, for videos in motion (such as aerial video) much of the data is redundant. The interesting information contained within the video are the spatial area covered in the scene and the dynamic activities (such as moving targets: cars, people, animals, etc) that can be used in the applications described above.

5. Visualizing Scenes: Approaches

In Section 3, we have shown a system that can take video sequences of large-scale scenes as input and is able to produce large field-of-view pushbroom mosaics, as well as reconstructed 3D data and dynamic object detection. In the following sections we describe how the multi-view mosaics and CB3M data are used and visualized now and how these may be delivered over the Internet.

5.1. Mosaic-based rendering

The first output we create after phase I is a set of multi-view pushbroom mosaics, as illustrated in Figure 3. We have applied our system to a minute long (HD – 1080p) aerial video of New York City. Figure 5 shows four close-up views of the (8 total) mosaics that were generated for the scene taken over New York City. The first mosaic close-up is created by one of the leading scanline's (forward looking relative to the motion) and the fourth mosaic close-up is created by a trailing scanline (backward looking). It can be visually inspected that the motion parallax of structures are preserved and the scene is aligned (note that these four mosaic close-ups represent novel parallel-perspective views from viewing angles that are far apart). There are two ways in which we can view the mosaics:

Mosaic viewing Viewing the mosaics independently already provides an efficient representation and summary of the scene being imaged. By stacking all of the mosaics generated and flipping among them we can observe that object movements and parallax are also preserved. We have thus efficiently represented a compressed 150MB, 1 minute long video in a set of 8 mosaics encoded in JPEG, with an average size of 2.5MB each. [23 – See nyc-2d.mov]

Stereo viewing The multi-view mosaics can also be rendered to view the 3D data using cyan-red glasses [23 – See nyc-anaglyph.mov] or shutter glasses. This is possible since the mosaics generated are aligned and in stereo correspondence. We have created a desktop application (Figure 7) that loads all of the mosaics and combines two views at a time to create an anaglyph by using the red channel from one mosaic and the green and blue channels of another. The application allows us to set the 'disparity' defined as the distance between the two consecutive mosaic views used, and the viewing angle by allowing the user to gradually change which two mosaics are being viewed (using viewing 'position' slider in Figure 7). In addition we can pan and zoom the mosaic. The application also works with shutter glasses and can be potentially used with polarized glasses.

Stereo viewing of multi-view mosaics works well and can be further improved by applying the techniques presented and evaluated by Idesis and Yaroslavsky [9].

Mosaic-based rendering can be used in Internet applications now since it only requires the use of color images. But with the current state of displays the user would have to use inexpensive cyan-red glasses or more expensive shutter glasses. The use of specialized glasses is cumbersome and causes viewers eyes to get tired. A better solution would be the use of 3D displays [ref. to Philips site] which would allow viewing of stereo mosaics without the need for specialized glasses. Currently 3D displays are rare and expensive.

5.2. Content-based 3D mosaic rendering

True 3D models can be visualized by rendering the 3D plane patches described in the CB3M reconstruction of scenes. In addition, it is possible to texture map the 3D models by using the multi-view pushbroom mosaics. On the desktop, scenes can be rendered using OpenGL or DirectX. A rendered 3D model will allow for additional information to be overlaid, such as images and videos, and hot spots that link to related websites or media.

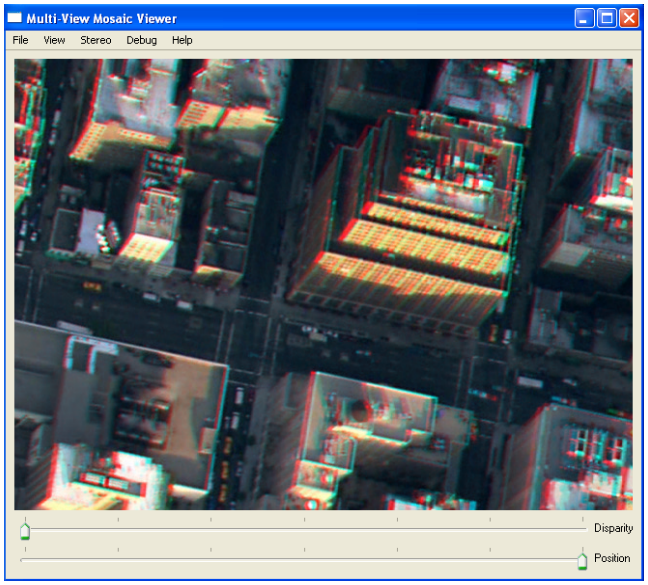

Figure 7: Screenshot of Multi-View Mosaic Viewer displaying a cyan-red anaglyph.

Rendering in true 3D over the web enables applications like those described above to be built. But while 3D formats for web delivery exist, 3D content is lacking, and web browser plug-ins and 3D browsers do not yet have a large user base. The CB3M format we use is an efficient format for storing and sharing large-scale 3D scenes, but is not meant to provide all of the additional features that standard formats provide. Although, it is possible to convert our CB3M data to other formats as necessary.

Various 3D formats exist for the Internet, many are proprietary or have evolved from products and others are open standards. Many websites and companies have used the formats but they still have not reached the market penetration other rich content formats on the web (such as Flash) have reached.

VRML and its successor X3D are both ISO standards and development is being led by the Web3D Consortium [19]. VRML (Virtual Reality Markup Language) was the first standard for 3D content on the Web and it has been widely used among 3D applications. X3D (which is based on XML) is the more recent ISO standard and successor to VRML. Many 3D applications can export to the VRML and X3D formats. Both VRML and X3D allow for scripting and animation of 3D models.

COLLADA is another XML based open format that is being developed as an industry standard for 3D data exchange. COLLADA's development is being led by the Khronos Group [3] which also controls many other open standards, which include OpenGL. The COLLADA specification was not designed for the web, but it can be used in conjunction with X3D [1] and provides many features for representing 3D data. Google Earth allows importing and exporting the COLLADA format in addition to its own KML format.

6. Conclusions and Discussions

This paper has presented a mosaic-based approach to large-scale scene modeling and rendering. Many interesting applications on the web would benefit and be created if many large-scale scenes could be modeled easily and inexpensively. We propose that by using video, much of which is already available, many cities and suburban areas could be modeled.

The approach as presented here works on aerial videos with a camera undergoing 6 degrees of freedom, but having an overall dominant direction of motion. Our lab is currently working on extending the geometry and mosaic-based approach to other typical motion trajectories such as circular camera motion and for more general camera motion. Work that we plan to do in the future include: handling occlusions for dynamic object tracking, and segmentation and indexing of the 3D models (currently the models are stored as a set of 3D planes in the CB3M format).

Formats and 3D web browsing applications continue to be developed but the content is currently lacking due to the manual process of creating 3D models. As automated reconstruction methods continue to be improved, 3D content will increase on the Web.

Although many videos are available on the Internet, it should be noted we have not discussed the issue of copyright since it is out of the scope of presenting the research proposed here.

7. Acknowledgements

This work is supported by AFRL/SN under Award No. FA8650-05-1-1853, by AFOSR under Award No. FA9550-08-1-0199, and by NSF under Grant No. CNS-0551598. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. However, the views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government.

8. References

[1] R. Arnaud, T. Parisi, Developing Web Applications with COLLADA and X3D. A Whitepaper. March 25, 2007.

[2] C. Baillard, A. Zisserman, Automatic reconstruction of piecewise planar models from multiple views, CVPR, vol. 2, 1999, pp. 559–565.

[3] COLLADA, http://www.khronos.org/collada/

[4] R. Collins, C. Jaynes, Y.-Q. Cheng, X. Wang, F. Stolle, E. Riseman, A. Hanson, The ascender system: automated site modeling from multiple aerial images, Comput. Vision Image Understand. 72 (2) (1998) 143–162.

[5] D. Comanicu, and P. Meer, Mean shift: a robust approach toward feature space analysis. PAMI, May 2002

[6] N. Cornelis, B. Leibe, K. Cornelis, L. V. Gool, 3D Urban Scene Modeling Integrating Recognition and Reconstruction. IJCV, Volume 78, July 2008.

[7] Google earth, http://earth.google.com/

[8] H. Tang, Z. Zhu, G. Wolberg and J. R. Layne, Dynamic 3D Urban Scene Modeling Using Multiple Pushbroom Mosaics, the *Third International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT 2006)*, University of North Carolina, Chapel Hill, USA, June 14-16, 2006.

[9] I. Ideses and L. P. Yaroslavsky, “Three Methods that improve the visual quality of colour anaglyphs”, 2005 J. Opt. A: Pure Appl. Opt. 7 755-762.

[10] Z. Kim, R. Nevatia, Automatic description of complex buildings from multiple images, Computer Vision and Image Understanding, 2004.

[11] R. Koenen, F. Pereira and L. Chiariglione, MPEG-4: context and objectives. Signal Processing: Image Communications, 9(4), 1997:295-300.

[12] Microsoft Virtual Earth, http://www.microsoft.com/virtualearth/

[13] M. Okutomi and T. Kanade, A multiple-baseline stereo, *PAMI*, vol. 15, no. 4, pp. 353-363. 1993

[14] M. Pollefeys, D. Nister, J.M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S.-J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewenius, R. Yang, G. Welch, H. Towles, Detailed Real-Time Urban 3D Reconstruction From Video. IJCV, Volume 78, July 2008.

[15] A. Rav-Acha, G. Engel, and S. Peleg, Minimal Aspect Distortion (MAD) Mosaicing of Long Scenes, IJCV, Volume 78, July 2008.

[16] I. Stamos, L. Liu, C. Chen,G. Wolberg, G. Yu, S. Zokai, Integrating Automated Range Registration with Multiview Geometry for the Photorealistic Modeling of Large-Scale Scenes. IJCV, Volume 78, July 2008.

[17] TerraSight 3D Visualizer, Sarnoff Corporation, http://www.pyramidvision.com/terrasight/tsite_1_content.html

[18] V. Verma, R. Kumar, S. Hsu, 3D Building Detection and Modeling from Aerial LIDAR Data, CVPR, vol. 2, 2006, pp. 2213 – 2220

[19] Web3D Consortium, http://www.web3d.org/

[20] W. Zhao, D. Nister, and S. Hsu, Alignment of Continuous Video onto 3D Point Clouds, CVPR, vol. 2, 2004, pp. 964 – 971

[21] Z. Zhu, T. Huang, Multimodal Surveillance: Sensors, Algorithms and Systems. Artech House Publisher, July 2007

[22] Z. Zhu, E. M. Riseman, A. R. Hanson, Generalized parallel-perspective stereo mosaics from airborne videos, PAMI, 26(2), Feb 2004

[23] http://visionlab.engr.ccny.cuny.edu/~molina/publications/08/nyc-videos.zip