Visualizing Very Large Image Data Sets At Interactive Rates

FRANK EDUGHOM EKPAR
Department of Computer Science
Admiralty University of Nigeria
Ibusa/Ogwashi-uku Expressway, Delta State
NIGERIA

HIROYUKI HASE
University of Fukui
Fukui City
JAPAN

MASAAKI YONEDA
Toyama National College of Technology
Toyama City
JAPAN

Abstract: - This paper presents a system for real-time visualization of very large image data sets using on-demand loading and dynamic view prediction. We use a robust image representation scheme for efficient adaptive rendering and introduce a fast perspective view generation module to extend the applicability of the system to panoramic images. We demonstrate the effectiveness of the system by applying it both to imagery that does not require perspective correction and to very large panoramic data sets requiring perspective view generation. Furthermore, we extend and generalize the system to enable a wide range of applications. In a broad set of applications, the system permits smooth, real-time interactive navigation of very large panoramic and non-panoramic image data sets on average personal computers without the use of specialized hardware.

Key-Words: - Big data, very large image data sets, multilayer image representation, dynamic view prediction, three-dimensional, multilayer electroencephalography (3D multilayer EEG).

Received: November 2, 2019. Revised: May 12, 2020. Accepted: May 18, 2020. Published: June 1, 2020.

1 Introduction

As a consequence of the decreasing cost and increasing capabilities of modern digital image capture, storage, manipulation and visualization hardware and software, the scope of applications of very large image data sets has expanded substantially in recent times. The creation, manipulation and rendering of medical image data sets has traditionally required specialized and relatively expensive computer systems due to the large size and complexity of the data. Demands for interactive visualization and data transmission over networks have further exacerbated the problem. Thomas A. Funkhouser and Carlo H. S'equin [1] describe an adaptive display algorithm for interactive frame rates during visualization of very complex virtual environments that relies upon a hierarchical model representation in which objects are described at multiple levels of detail and can be drawn with various rendering algorithms. Funkhouser et al adjust image quality adaptively to maintain a uniform, user-specified target frame rate. This solution often permits real-time visualization at the expense of image quality, especially when data is transmitted over bandwidth-limited networks. Ron Kikinis et al [2] developed a digital brain atlas for surgical planning, model driven segmentation and teaching comprising a three-dimensional (3D) digitized atlas of the human brain to visualize spatially complex structures that was designed for use with magnetic resonance (MR) imaging data sets. Reliance on expensive computer systems, hardware rendering and the use of pre-rendered views on inexpensive computer systems limited the applicability of this solution. The brain atlas was later extended with a multi-layer image representation and compression scheme designed to allow average users to interact with high-quality 3D scenes derived from large data sets typically found in medical applications. This extension did not directly support real-time navigation or multi-resolution zoom capabilities but allowed other types of interactivity such as turning structures on and off, and adjusting their transparency and color. Solutions to the problems of rendering and effects in large-scale image data sets have been described in [3], [4] and [5].

The use of panoramic imaging techniques dates back as far as the 18 th century [6] when early works of art featuring panoramic projections were created. A wide variety of techniques for creating, storing, transmitting and visualizing panoramic images have been reported in the scientific literature. One key challenge for a panoramic imaging system is the encoding of enough visual information to make the system practical and cost-effective. The use of rotating cameras to build panoramic mosaics, as described in [7] and [8], while facilitating the capture of panoramas of static scenes, is of limited practical value in more dynamic scenes. Gregus [9] introduced a compact omnidirectional image capture system comprising both reflective and refractive elements (catadioptric) and that is capable of capturing a complete 360-degree field of view in a single image frame with no moving parts. Since they generally have no moving parts and can generate a panoramic view in a single image frame, catadioptric systems can be used to capture both static and dynamic scenes and video in real-time. Systems with moving parts or systems that need to be rotated to capture a panorama cannot be used to capture dynamic scenes because after capturing one part of a scene and then moving or rotating to capture another part, the parts of the scene will be out of sync in time since the rotation or movement takes time. Catadioptric systems have no moving parts and do not need rotation so they can capture the entire scene at once – thus permitting the capture of both static and dynamic scenes. The reflective elements in these systems can be designed to capture the entire 360- degree field of view and a vertical field of view approaching 180 degrees with relatively low levels of distortion. The light from the reflective elements can then be focused by the refractive elements for image formation. Aspects of the geometrical characteristics of catadioptric systems were described by Geyer et al [10]. Yagi et al [11] and Chahl et al [12] proposed other panoramic image capture systems that are amenable to real-time operation. The popular Apple QuickTime Virtual Reality Authoring System [13] allows the generation of cylindrical panoramic images from several overlapping still image segments captured using a conventional camera and provides rendering and multimedia integration features. Robust methods for navigating panoramic images and for correcting distortions in panoramic images have been introduced by Frank et al in [14] and [15].

Existing methods for the visualization of panoramic images on average personal computers generally apply to relatively small and medium scale images. This is because the personal computers have main memory in the range of gigabytes so only a few frames can fit in memory at a time, necessitating the loading of images from disk and making real-time operation difficult or impossible. We present techniques for the visualization of very large (in the order of several gigapixels or more per image frame) panoramic and non-panoramic images at real-time frame rates on average personal computers without the use of specialized hardware. Our system uses a robust image representation scheme for efficient adaptive rendering. This image representation is explained in greater detail in Section 3 and it permits the application of the dynamic view prediction technique described in Section 4. We also introduce a fast set of perspective view generation equations for the manipulation of panoramic images.

This paper is an extended version of earlier results presented by the authors at an IEEE Conference [18] and improves the original paper by introducing a generalized formulation of the techniques presented in this paper to a wide range of application domains. Furthermore, the generalization enables intelligent navigation of the vast amounts of data that could be generated by the three-dimensional multilayer electroencephalography systems that are a subset of the systems described by F. E. Ekpar [19]. The major contributions of this work are the introduction of dynamic view prediction, the multi-tier image representation scheme and the generalization of the system to a wide variety of application domains. Research by Vinegoni C. et al [20] has led to the development of a visualization technique that enables rapid, multiscale, analysis and virtualization of large-volume, high-resolution complex biological data while generating 3D tractographic representations. Our system can be applied to this technique to enable real-time visualization of very large data on general- purpose systems. Other situations in which our system can be applied include the systems presented by Luigi Gallo et al [21] for high-fidelity visualization of large medical datasets and the V3D system by Hanchuan Peng et al [22].

The remainder of this paper is organized as follows: Section 2 introduces the generalized formulation for the interactive visualization of big data including very large image data sets. Section 3 describes the image representation schemes we have applied to the problem. The interactive visualization process is explained in Section 3. Section 4 describes an application of the proposed solution to non-panoramic image data while Section 5 describes the visualization of very large panoramic images using the techniques presented in this paper. Concluding remarks appear in Section 6.

2 Generalized Formulation

Efficient data management techniques such as the use of predictive loading of relevant data and possible subsequent presentation on a display window or computer monitor or any other suitable device or system based on a dynamic prediction of the user's point of view within the data stream could be applied to enable practical implementation and acceptable performance of applications of the system for very large data sets and related applications on off-the-shelf personal computer systems. When data associated with any selected instance is characterized in the form of pixel values of images representing the data, dynamic view prediction of the user's field of view and associated intelligent data management techniques could be used to facilitate efficient data navigation.

Generally, any data set including, but not limited to very large data sets or image data sets, could be navigated seamlessly even in resource-limited environments such as the Internet by presenting a user-selected region of interest and permitting real-time operation by dynamically predicting and making available subsets of the original data set.

Equations 1, 2, 3 and 4 summarize aspects of the general principles of operation for the seamless navigation of data sets including, but not limited to, very large data sets or image data sets.

$$R^K = f\left(X^N\right) \quad (1)$$

$$X^N = \{x_1, x_2, \ldots, x_N\} \quad (2)$$

$$\{T, \theta, \varphi, r, \lambda\} \subset X^N \quad (3)$$

$$\{B_{11}, \ldots.., B_{NM}\} \subset R^K \quad (4)$$

In Equation 1, R^K denotes the region of interest within the original data set characterized by a set of K features. The features could be a set of pixels in a 2D or 3D still or time-varying image, a set of image blocks (possibly at a given level of a resolution-based hierarchical pyramid) within a 2D or 3D still or time-varying image, or any other suitable feature set within an appropriate data set. X^N is a set of N parameters (decomposed into the component parameters $\{x_1, x_2, \ldots, x_N\}$ in Equation 2) that determine the characteristics of the region of interest. As indicated in Equation 1, the region of interest is a function of the parameter set X^N.

Note that for sufficiently small data sets, the region of interest could encompass the entire data set.

For image data, the observation that interactive rendering of the image data set involves the display of a relatively small (compared to the size of the underlying image data itself) view window could be exploited to adopt a robust two-tier or bi-level representation of the image. The first level would contain a virtual view of an entire image frame as a single continuous set of pixels. Note that the actual pixels – which in reality could be representations of entities other than images – need not be associated with a single image or image frame but could be considered as a continuous collection or set for simplicity in management. The region of interest or view window could then be calculated using a suitable application-specific feature set such as the time-varying 3D spherical coordinate feature set depicted in Equation 3 where T represents time, θ represents the pan angle, φ represents the tilt angle or azimuth, r represents the radius of the sphere while λ represents a zoom level that could be used to control the amount of detail for each view – as a subset of the total parameter set X^N. Since a single image frame can be very large, it is generally impractical to attempt to load the entire image frame (typically corresponding to tens of gigabytes or even terabytes or more of physical memory for certain applications) into memory at once. Consequently, the second level could comprise a segmentation or partitioning of each image frame into distinct image blocks of a size and color depth that facilitates straightforward manipulation on an average personal computer. This partitioning schem e could be accomplished by segmenting each i mage frame into NxM distinct im age blocks labeled B_{11}, B_{12} ,…, B_{NM} as illustrated in Equation 4 where the set of image blocks is depicted as a subset of the region of interest.

3 Image Representation

Based on the observation that interactive rendering of the im age data set involves the display of a relatively small (compared to the size of the underlying image data itself) view window, we use a robust two-tier or bi-level representation of the image. The first level cont ains a virtual view of an entire image frame as a single continuous set of pixels. Figure 1 illustrates the first level for a two-dimensional image frame of width P_W and height P_H pixels. The region of inter est or view window is indicated as V in Fig. 1. Since a sin gle image frame can be ver y large, i t is generally im practical to attempt to load the entire image frame (corresponding to tens of gigabytes of physical memory) into mem ory at once. As explained in Section 2, the seco nd level co mprises a segmentation or partitioning of each i mage frame into distinct image blocks of a size and color depth that facilitates straightforward manipulation on an average personal com puter. This partitioning scheme is sh own in Fig. 2, where the image of Fig. 1 has been segmented into NxM distinct image blocks labeled $B_{11}, B_{12}, \ldots, B_{NM}$. In summary, the two-tier representation permits us to view the entire image as a single image in the first tier and as a s et of blocks that can be loaded on demand in the second tier. The size of each partition ca n be chosen such that the view window, V , straddles just a couple of i mage blocks. In this case onl y those image blocks in the second layer that are covered or straddled by the view win dow need be loaded into memory for the manipulation or rend ering of the view, leading to a significantly reduced memory footprint.

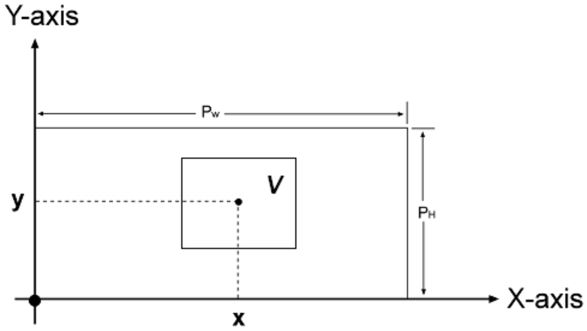

Fig. 1: Tier-1 image representation

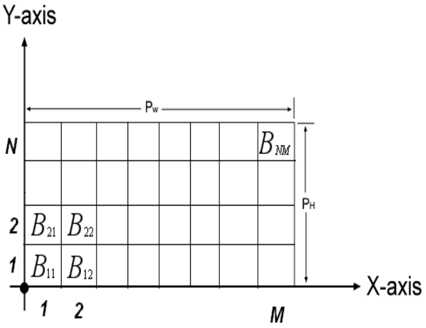

Fig. 2: Tier-2 image representation

The use of a two-tier i mage representation scheme permits alternate views of the image data that mak e further manipulation easier. For exam ple, the simplicity of the first level permits the application of a multi-resolution pyramid representation of the image data, such as that described by Peter J. Burt et al. [16], for efficient compression, storage and transmission and optio nally for adapti ve rendering that maintains a constant fram e rate. A thumbnail of the entire i mage could als o be generate d at the fir st level. Such a thum bnail could be used to displa y a lower resolution version of the view wi ndow while waiting for image data to be retrieved and/or decompressed. Furthermore, the dyna mic view prediction and on-demand loading algorithms described hereinafter are readily applicable to the second tier’s image block representation.

4 Interactive Visualization Process

We now outl ine the proce ss of visualizing the data sets. First, a view window V is specified as illustrated in Fig. 1. The view window represents the segment of the current image frame that is indicated by the view parameters. In the current implementation, three view parameters are used to control the view. These are the pan angle (θ), the tilt angle or azimuth (φ) and the zoom or scale factor. User input is received via the keyboard and/or mouse clicks within the view window. Views are generated based on view window size and received input. In order to facilitate interactive rendering, the rate of change of each of the view parameters with respect to time is computed dynamically. The computed rate of change is then used to predict the value of the parameter at any desired time in the future. Equation 5 illustrates the use of the dynamic view prediction algorithm for a specific view parameter P -- pan, tilt or zoom level.

$$p = p_0 + KaT \qquad (5)$$

In Equation 5, P is the predicted value of the parameter at time T , p_0 is the current value of the parameter, a is the dynamically computed rate of change of the parameter with respect to time and K is a scale factor, usually 1. The values of the parameters predicted by Equation 5 are used to determine which specific image blocks need to be loaded into memory at any given time. Our implementation uses a background thread dedicated to loading those image blocks that are covered by the current view as well as any additional image blocks that might be needed for rendering the view in the future, that is, a number of future time steps. Since the number of image blocks per frame is usually small, it is practical to preload image blocks

In Equation 5, P is the predicted value of the parameter at time T , p_0 is the current value of the parameter, a is the dynamically computed rate of change of the parameter with respect to time and K is a scale factor, usually 1. The values of the parameters predicted by Equation 5 are used to determine which specific image blocks need to be loaded into memory at any given time. Our implementation uses a background thread dedicated to loading those image blocks that are covered by the current view as well as any additional image blocks that might be needed for rendering the view in the future, that is, a number of future time steps. Since the number of image blocks per frame is usually small, it is practical to preload image blocks

that would be required for rendering several time steps ahead – permitting smooth rendering at real-time rates.

5 Application to Large Non-panoramic Image Data Sets

First, we apply the techniques described herein to large non-panoramic data sets. Let us consider the image shown in Fig. 1. For the sake of simplicity and to facilitate uniformity with panoramic image data sets, we represent pixel positions on the image as angular displacements. Each position along the horizontal axis is represented by a corresponding angle θ between 0 and 2π radians while each position along the vertical axis is represented by a corresponding angle φ between $-\pi/2$ and $\pi/2$ radians. Using this representation, Equations 6 and 7 depict how the required angular coordinates θ and φ can be obtained for the point with coordinates x and y in Fig. 1.

$$\theta = \frac{2\pi x}{P_W} \qquad (6)$$

$$\varphi = \frac{\pi y}{P_H} - \frac{\pi}{2} \qquad (7)$$

5.1 Image blending

Although in principle it is possible to capture very large images of the type described here in a single image frame using a single imaging device, it is often more practical to capture a series of overlapping high-resolution segments of the image and then stitch these together to form a single image mosaic. Techniques for stitching overlapping image segments are well known. One significant problem with image stitching is how to make the seams between overlapping segments invisible. A wide variety of image blending techniques exist. Generally, the choice of a specific image blending technique depends on the requirements of the specific application. The multi-resolution spline technique proposed by Peter J. Burt et al [17] gives satisfactory results albeit requiring large amounts of memory. Peter J. Burt et al first decompose the images to be splined into a set of band-pass filtered component images with the component images in each frequency band assembled into a corresponding band-pass mosaic. Component images are joined using a weighted average within a transition zone proportional in size to the wavelengths that comprise the band.

Ultimately, summation of the band-pass mosaic images is used to derive the output mosaic image.

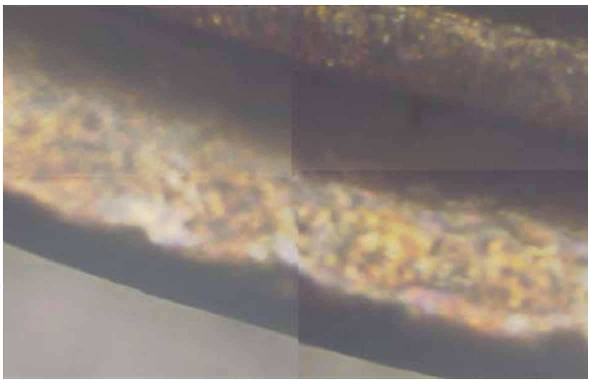

Figure 3: Image segments stitched together without blending

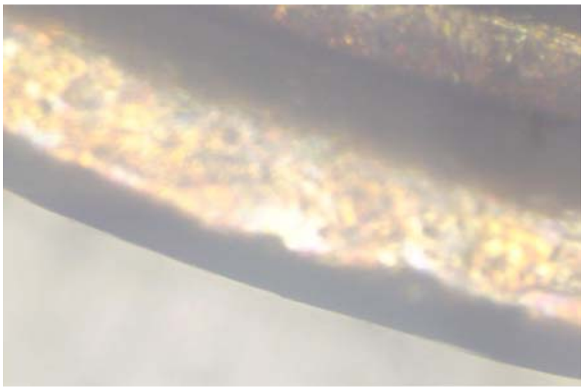

Figure 4: Image segments of Figure 3 stitched together by applying multi-resolution spline blending

We have used the image representation scheme described here to efficiently store and manipulate the sub-images required by the multi-resolution spline technique. Fig. 3 shows part of an image mosaic obtained by stitching overlapping image segments without the application of image blending. Note the visible seams in the image. Fig. 4 illustrates the same image with the multi-resolution spline technique used to blend the overlapping regions together. As Fig. 4 shows, this technique effectively removes the seams.

5.2 Navigating non-panoramic images

Image navigation for non-panoramic data sets comprises the display of the selected portion of the image data set. User input could be obtained using standard input devices such as the mouse, keyboard and joystick. Simple image transformations such as zooming, rotation could be applied to the views.

6 Application to Panoramic Images

Panoramic images can be acquired using a wide variety of techniques. One of the simplest methods of acquiring panoramic images involves the capture of a series of overlapping image segments around a unique effective viewpoint using a conventional digital camera.

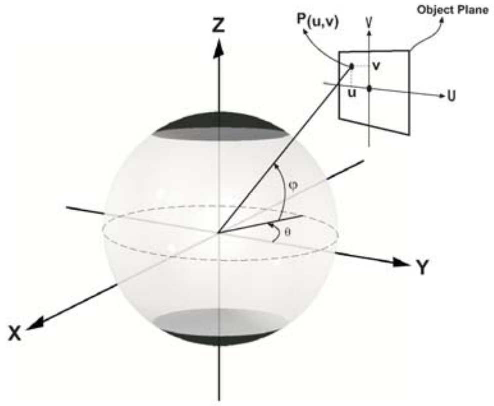

Fig. 5: Image representation on a spherical surface

The overlapping segments are then stitched together and blended (using the multi-resolution spline technique [17] or any other appropriate technique) to produce the complete panoramic mosaic.

6.1 Spherical Projection

Panoramic images are first re-projected onto a spherical surface. This allows us to apply a uniform perspective correction algorithm during the navigation of the image. It is convenient to realize the spherical representation in the first tier of the image data set representation we have proposed here. Fig. 5 is a conceptual illustration of the projection or re-projection of a panoramic image onto the surface of a sphere. Portions of the sphere for which panoramic image data is not available (for example, when the panorama is acquired using a system with a vertical field of view that is less than π radians) can be replaced with user-supplied data or simply filled with a uniform color. The region of interest or view window (corresponding to V in Figure 1) is depicted on the $U-V$ coordinate plane as a perspective projection of the region of the panorama indicated by the viewing parameters.

6.2 Catadioptric system application

Let us apply the techniques proposed in this paper to panoramic images acquired using a catadioptric system. Fig. 6 is a schematic diagram illustrating image formation by the panoramic annular lens (PAL) catadioptric panoramic imaging system [9] while Fig. 7 shows a sample PAL system. In Figure 6, φ_A represents the limit of the vertical field of view above the horizon while φ_B represents the limit of the vertical field of view below the horizon. For simplicity, let us assign a positive value to φ_A and a negative value to φ_B.

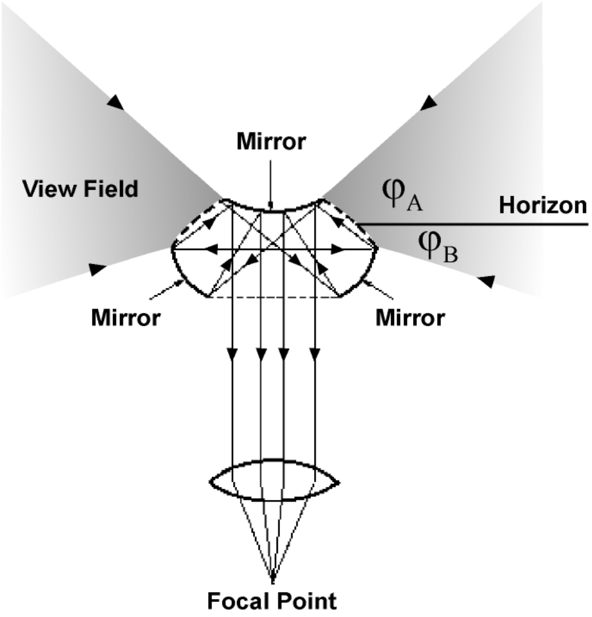

Fig. 6: Schematic diagram of catadioptric system

Fig. 7: Sample catadioptric system

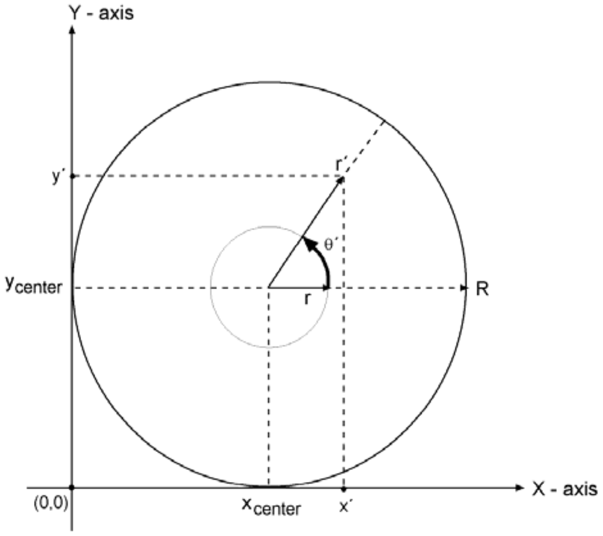

Fig. 8: Image formation by catadioptric system of Fig. 6

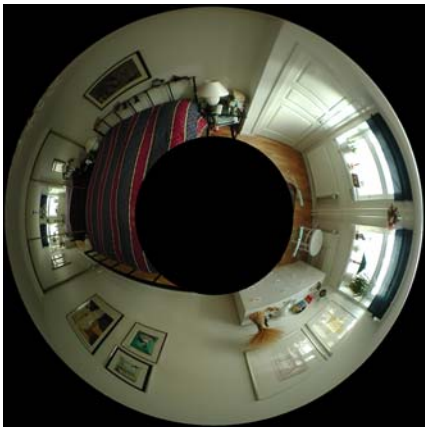

Fig. 9: Actual image of bedroom captured using catadioptric system of Fig. 6

The image formed by the PAL system is donut shaped. Lines perpendicular to the optical axis of the PAL system are projected onto concentric circles in the PAL image while lines parallel the optical axis are projected onto radial lines. Fig. 8 is a conceptual representation of a PAL image while Fig. 9 shows an actual image of a bedroom captured with the PAL system. In Fig. 8, R is the outer radius while r is the inner radius of the PAL image. The center of the PAL image is marked as the point

(x_{center}, y_{center}). Equations 8 and 9 describe the relationship between the Cartesian coordinates of a point (x', y') and its polar coordinates (r', θ').

$$x' = x_{center} + r' \cos\theta' \tag{8}$$

$$y' = y_{center} + r' \sin\theta' \tag{9}$$

The coordinate transformation shown in Equations 8 and 9 could be used to extract a rectilinear 360-degree panoramic image from the donut-shaped PAL image. Fig. 10 shows a 360-degree panoramic image obtained from a donut-shaped PAL image using Equations 8 and 9.

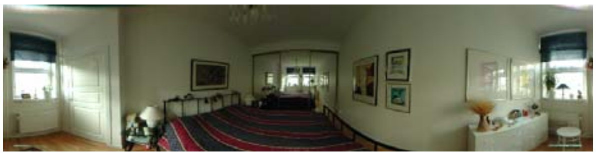

Fig. 10: Flattened 360-degree panoramic image obtained from a donut-shaped PAL image of Fig. 9

For the purpose of representing the PAL image on a spherical surface, the required θ (between 0 and 2π radians) and φ (between φ_B and φ_A) angles are obtained from the PAL image (refer to Fig. 8) as indicated in Equations 10 and 11.

$$\theta = \theta' \tag{10}$$

$$\varphi = \varphi_B + \left(\frac{r'}{R-r}\right)\left(\varphi_A - \varphi_B\right) \tag{11}$$

6.3 Perspective correction

In addition to the correction of any substantial geometric distortions, views extracted from panoramic images require perspective correction for comfortable viewing by a human observer. Figure 5 illustrates the generation of a perspective-corrected view and its representation on the $U-V$ coordinate plane. With each input image frame represented on the surface of a sphere, we use a simplified set of equations to construct the perspective-corrected view. Suppose that the ray from the center of the sphere to the center of the $U-V$ coordinate plane (the viewing direction for the indicated view) intersects the surface of the sphere at the point with angular coordinates (θ_0, φ_0), then the point with coordinates (u, v) on the perspective-corrected $U-V$ image plane corresponds to the point with coordinates (θ, φ) on the surface of the sphere. The perspective correction is accomplished as described in Equations 12 and 9.

$$\theta = \theta_0 + a\tan2(u,(Z\cos\varphi_0 - v\sin\varphi_0)) \tag{12}$$

$$\varphi = \varphi_0 + a\sin\left(\frac{Z\sin\varphi_0 + v\cos\varphi_0}{\sqrt{Z^2+u^2+v^2}}\right) \tag{13}$$

In Equations 12 and 13, Z is an arbitrarily chosen zoom factor. Although the perspective correction described in Equations 12 and 13 can be implemented in software on an average personal computer, our experiments have shown that speed increases of several orders of magnitude can easily be obtained without visibly sacrificing image quality by partitioning the $U-V$ image plane into a grid of control points and then applying the perspective correction to the control points only, with points in-between estimated using simple bilinear interpolation. As explained in Section 6.1, it is advantageous to realize the spherical representation of the panoramic image in the first tier of the image data set. Those image blocks in the second tier that are indicated by the subset of coordinates on the surface of the sphere corresponding to the selected view are loaded into memory and the dynamic view prediction algorithm described in Section 4 is applied to facilitate fast rendering.

Fig. 11 shows a portion of a 360-degree panoramic image of an office with visible perspective distortion at the top of the window and ceiling. In Fig. 12, the perspective correction algorithm described here has been applied to the image of Fig. 11.

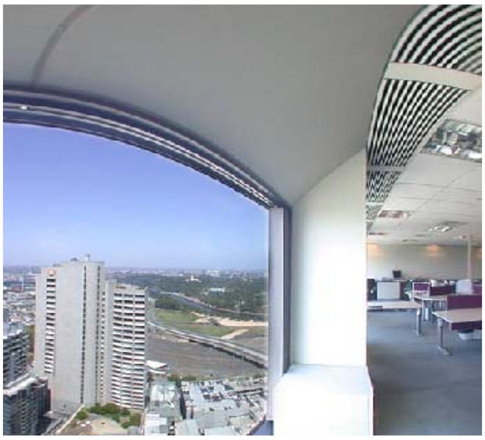

Fig. 11: Portion of 360-degree panoramic image with visible perspective distortion

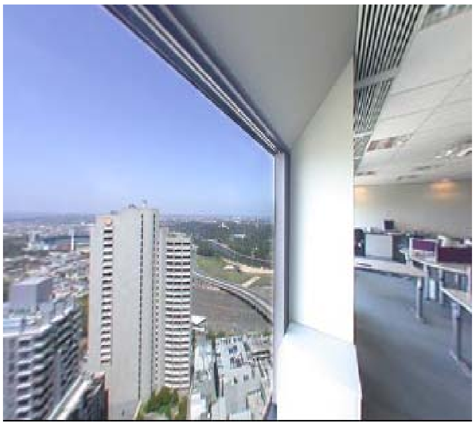

Fig. 12: Image of Fig. 11 after perspective correction

7 Conclusion

In this paper, we described a system for the real-time visualization of very large image data sets using on-demand loading and dynamic view prediction. We used a two-tier representation of images that is amenable to efficient adaptive rendering and combined it with dynamic view prediction and on-demand loading as well as a generalization of the techniques to enable application across a wide variety of contexts. As a demonstration of the effective of the system we proposed, we described two applications – one for panoramic images and the other for non-panoramic images. Our experimental results indicate that the system is capable of permitting smooth, real-time rendering of very large images on average personal computers without the use of specialized hardware. The system can be used with data transmitted over a network with the two-tier representation allowing selection of an appropriate representation of the data to match network bandwidth and traffic conditions. We have also shown how to extend the range of applications of the system by generalizing the techniques presented. In the future, we hope to study the effect of noise in very large images.

References:

[1] Thomas A. Funkhouser and Carlo H., “Adaptive Display Algorithm for Interactive Frame Rates During Visualization of Complex Virtual Environments”, *Proceedings SIGGRAPH*, 1993, pp. 247-254.

[2] R. Kikinis, M. E. Shenton, D. V. Iosifescu, R. W. McCarley, P. Saiviroonporn, H. H. Hokama, A. Robatino, D. Metcalf, C. G. Wible, C. M. Portas, R. M. Donnino, and F. A. Jolesz, “A digital brain atlas for surgical planning, model driven segmentation and teaching”, *IEEE Transactions on Visualization and Computer Graphics*, Volume 2, Issue 3, 1996, pp. 232-241.

[3] D. Laur and P. Hanrahan, “Hierarchical splatting: A progressive refinement algorithm for volume rendering”, *ACM SIGGRAPH Computer Graphics Proceedings*, 25(4), 1991, pp. 285-288.

[4] Marc Levo, “Efficient ray tracing of volume data”, *ACM Transactions on Graphics*, 9(3), 1990, pp. 245-261.

[5] Abraham Mammen, “Transparency and antialiasing algorithms implemented with the virtual pixel maps technique”, *IEEE Computer Graphics and Applications*, 9(4), 1989, pp. 43-55.

[6] R. Benosman, “Panoramic imaging from 1767 to the present”, *Proc. International Conference on Advanced Robotics, Workshop*, Volume 1, 2001, pp. 9-10.

[7] R. Szeliski, “Video mosaics for virtual environments”, *Computer Graphics and Applications*, 16(3), 1996, pp. 23-30.

[8] H. Sawhney, S. Hsu, and R. Kumar, “Robust video mosaicing through topology inference and local to global alignment”, *Proc. Fifth European Conference on Computer Vision*, Volume 2, 1998, pp. 103-119.

[9] P. Greguss, “PAL-optic based instruments for space research and robotics ”, *Laser and Optoelektronik*, Volume 28, 1996. pp. 43-49.

[10] C. Geyer, and K. Daniilidis, “Catadioptric projective geometry”, *Proc. International*

Conference on Advanced Robotics, Volume 1, 2001, pp. 17-30.

[11] Y. Yagi, S. Kawato, and S. Tsuji, "Real-time omni directional image sensor (copis) for vision-guided navigation*", IEEE Transactions on Robotics and Automation*, 10 (1), 1994, pp. 11-22.

[12] J. Chahl, and M. Srinivasan, "Reflective surfaces for panoramic imaging", *Applied Optics*, Volume 36, 1997, pp. 8275-8285.

[13] Shenchang Eric Chen, "QuickTime VR: an image-based approach to virtual environment navigation", *Computer Graphics (Proc. SIGGRAPH)*, 1995, pp. 29-38.

[14] F. Ekpar, H. Hase, and M. Yoneda, "Constructing arbitrary perspective -corrected views from panoramic images using neural networks", *Proc. 7th International Conference on Neural Information Processing*, Volume 1, 2000, pp. 156-160.

[15] F. Ekpar, H. Hase, and M. Yoneda, "Correcting distortions in panoramic images using constructive neural networks", *International Journal of Neural Systems,* Volume 1, 2003, pp. 239-250.

[16] Peter J. Burt and Edward H. Adelson, "The Laplacian Pyramid as a Compact Image Code", *IEEE Transactions on COmmunications*, 1983, pp. 532-540.

[17] Peter J. Burt and Edward H. Adelson, "A Multiresolution Spline with Application to Image Mosaics", *ACM Transactions on Graphics,* Volume 2, Number 4, 1983, pp. 217-236.

[18] Frank Ekpar, Hiroyuki Hase and Masaaki Yoneda, On the Interactive Visualization of Very Large Image Data Sets, *Proceedings of the 7th IEEE Conference on Computer and Information Technology*, 2007, pp. 627-632.

[19] Frank E. Ekpar, System for Nature-inspired Signal Processing: Principles and Practice, *European Journal of Electrical Engineering and Computer Science*, Vol.3, No.6, 2019: 1-10.

[20] Vinegoni, C., Fumene Feruglio, P., Courties, G. *et al.*, Fluorescence microscopy tensor imaging representations for large-scale dataset analysis, *Scientific Reports*, 10, Article Number 5632, (2020).

[21] Luigi Gallo, Alessio Pierluigi Placitelli, High-Fidelity Visualization of Large Medical Datasets on Commodity Hardware, *ISRN Biomedical Engineering*, Vol.2013, Article ID 892967, (2013).

[22] Hanchuan Peng, Zongcai Ruan, Fuhui Long, Julie H. Simpson, Eugene W. Myers, V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets, *Nature Biotechnology*, Vol.28, No.4, 2010: 348–353.