Visualizing 3D Scenes using Non-Linear Projections and Data Mining of Previous Camera Movements

Karan Singh, Ravin Balakrishnan
Department of Computer Science
University of Toronto
karan | ravin @ dgp.toronto.edu

ABSTRACT

We describe techniques for exploring 3D scenes by combining non-linear projections with the interactive data mining of camera navigations from previous explorations. Our approach is motivated by two key observations: First, that there is a wealth of information in prior explorations of a scene that can assist in future presentations of the same scene. Second, current linear perspective camera models produce images that are too limited to adequately capture the complexity of many 3D scenes. The contributions of this paper are two-fold. First, we show how spatial and temporal subdivision schemes can be used to store camera navigation information that is data mined and clustered to be interactively applicable to a number of existing techniques. Second, we show how the movement of a traditional linear perspective camera is closely tied to non-linear projections that combine space and time. As a result, we present a coherent system where the navigation of a conventional camera is data mined to provide both the understandability of linear perspective and the flexibility of non-linear projection of a 3D scene in real-time. Our system's generality is illustrated by three visualization techniques built with a single data mining and projection infrastructure.

Categories and Subject Descriptors

I.3.3 [**Computer Graphics**]: Picture/Image Generation – *viewing algorithms;* I.3.6 [**Computer Graphics**]: Methodology and Techniques – *interaction techniques.*

General Terms

Algorithms, Design, Human Factors

Keywords

Non linear projection, data mining, camera visualization

1. INTRODUCTION

Complex 3D models and scenes can now be realistically rendered in real-time on commodity hardware. However, interactive viewing, navigation, and presentation of complex 3D scenes remain difficult to achieve effectively. There are a number of reasons for this but two are of particular relevance to this paper. First, interaction with virtual 3D scenes, given current 2D display technology, is typically achieved through 2D projections, which inhibits a viewer's overall spatial understanding of the scene. Second, navigation through a virtual 3D scene offers viewing possibilities which transcend those of a real physical scene.

Without good exploration and visualization tools it is easy for a viewer to "get lost" in a scene, view models from awkward angles, miss important features, and experience frustration at their inability to navigate as desired. In some cases, 2D imagery can be more effective in conveying information about the 3D scene. On the other hand, static 2D linear perspective imagery of a 3D scene is disjointed and gives no feedback on a viewer's degree of interest in different parts of a scene. While linear perspective – which is the simplest 2D projection that provides information about the third dimension via easily understood depth cues – provides an easily understood method for exploring and visualizing localized regions of a model, it can be restrictive for the visualization of complex scenes. Artists working in traditional media have used non-linear projections effectively for centuries to express 3D shapes in 2D imagery but its use for interactive computer visualization is relatively new [29]. Also, most current visualization tools do not exploit the wealth of information that can be gleaned from prior explorations of a scene.

The goal of our work is to allow artists, designers, and end viewers to leverage off past and ongoing camera manipulation to help explore, understand, and subsequently express 3D scenes using non-linear projective imagery. To motivate this objective, consider the following three examples, illustrated by Figures 1-3, which effectively utilize non-linear projection and information from previous camera manipulations.

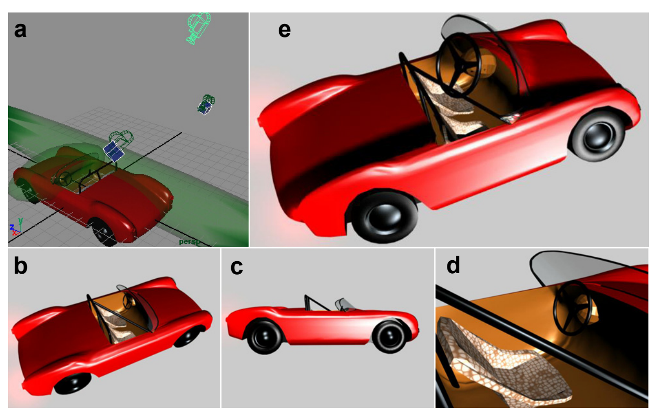

Figure 1. Mosaic camera. a) Schematic of three cameras viewing a scene. Mosaic camera is in bright-green, and two other cameras have viewports in blue and focus in green. b) Linear perspective through mosaic camera. c,d) View through other two cameras. e) Resulting non-linear projection combining views b,c,d, as viewed through mosaic camera.

Afrigraph'04, November 3–5, 2004, Cape Town, South Africa

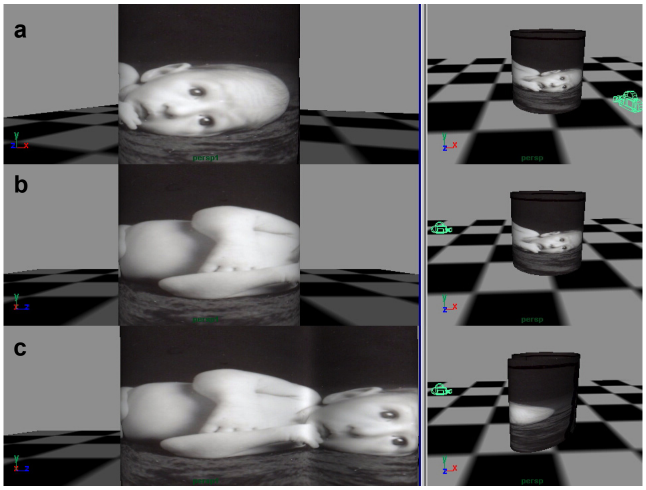

Figure 2. ***Sticky camera*: images in the left column show views through the lens, while the right column shows the respective locations of the camera in the scene. a,b) Regular linear perspective view as the camera is moved quickly from 2a to 2b. c) The camera data captured in the movement from 2a to 2b is used to create a non-linear projection, resulting in a "sticky" camera. On the right, the scene is seen distorted such that it projects correctly through the sticky camera's lens.**

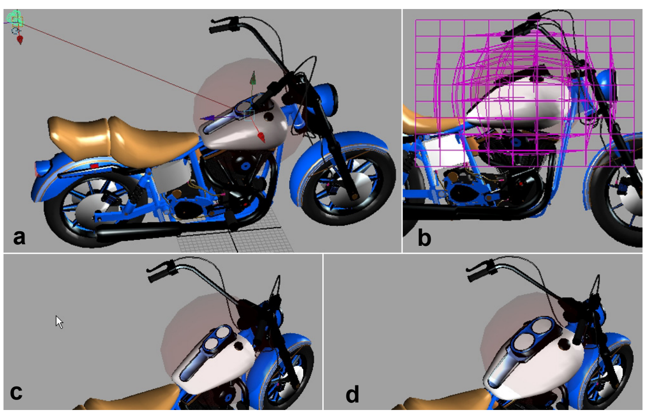

Figure 3. ***Fisheye Camera*: a) This global view shows the camera, and a spherical radial falloff function around its center of interest. b) Rectilinear grid surrounds center of interest. c) Through-the-lens linear perspective view of the scene. d) Resulting fisheye view after some time has been spent focusing on the gas tank of the motorcycle.**

The first example is the *Mosaic camera* (Figure 1). Three different views of a car (Fig 1b-d), from different cameras (Fig 1a), are combined into a single non-linear projection (Fig 1e) where the overall shape of the car is presented. In the same image, the region around the steering wheel is zoomed in (from view in 1d), to highlight the styling of the interior. Also, the side panel on the left is rotated outwards to show the wheels and character lines along the side of the car (from 1c). These three views were automatically extracted by data mining previous explorations of the scene for views where users dwelt for long periods of time. They were then combined into a single non-linear projection of the scene. This example illustrates how non-linear projections that possess a local linear perspective are flexible, while still being easily deciphered by a viewer.

The second example is the *Sticky camera* (Figure 2). One often moves the camera to view a part of the scene currently obscured at the expense of regions currently in view. A sticky camera, while moving to bring new parts of the scene into focus, would use a non-linear projective warp to prevent parts of the scene previously in view from disappearing, in proportion to the time spent focusing them. The sticky camera can be thought of as combining the views from a trail of cameras behind it whose influence fade with time. This is seen in Figure 2 where the face that was in focus in 2a, disappears in 2b as the camera orbits the model. The sticky camera in 2c, however, keeps the face in view, while focusing on the torso as in 2b.

The third example is the *Fisheye camera* (Figure 3), where the amount of time spent focused on different parts of a scene is proportional to its size in any given view. Here as in Figure 2, the concepts of viscoelasticity of a view is directly applied.

The contribution of our research is the concept and development of an infrastructure within which complex non-linear visualizations of scenes can be built up from previous navigations using conventional linear cameras. Our approach is motivated by the boot-strap nature of the presentation of a 3D scene, in that an author of a viewing experience must first explore the scene as a viewer, likely resulting in a strong relationship between exploration and presentation of a scene. It is these relationships that we data mine to enhance 3D scene visualization through appropriate non-linear projections.

The rest of this paper is organized as follows: Section 2 reviews prior art. Section 3 proposes efficient spatial and temporal data structures for storing and processing camera attributes for subsequent data mining. Section 4 discusses details of how this infrastructure is used to generate three new visualization techniques: the *Mosaic, Sticky,* and *Fisheye* cameras (Figures 1-3). Section 5 concludes with a discussion of the results obtained.

2. RELATED WORK

2.1 Camera Navigation

A variety of metaphors have been developed to assist the user when navigating 3D virtual environments. One of the most common metaphors, the cinematic camera, allows users to track, tumble, and dolly the camera around a scene. Other metaphors include through-the-lens control [11], flying and orbiting [31], points and areas of interests [17], path drawing [15], bimanual techniques [36], using constraints [21], and various combinations of techniques [30, 35]. Taxonomies and evaluations of these various techniques are presented by Bowman et al. [2]. Our current paper presents techniques for data mining camera explorations that use any of the above camera manipulations.

Apart from direct techniques for navigating the scene, additional information can be provided to aid navigation. These include landmarks [6], and global overview maps in addition to local views [8]. The integration of global and local views, using various distorted spaces including "fisheye" views has also been explored [4, 9]. Our present work uses the data mining of camera manipulation to construct such visualizations.

Another approach is to give the author more influence over what the viewer sees by creating guided tours where camera paths are prespecified to varying degrees. Hanson and Wernert [13, 32] propose "virtual sidewalks" which are authored by constructing virtual surfaces and specifying gaze direction, procedural events, and vistas along a sidewalk. Galyean [10] proposes a "river analogy" where a user, on a metaphorical boat, can steer the boat in limited deviations from the prescribed "river" path. Automatic creation of paths using robotic planning algorithms have also been investigated [7]. Automatic camera framing of a scene by following defined rules such as keeping certain objects visible in the scene, or following a primary object have also been explored [14]. Burtnyk et al. [3] describe the StyleCam system that allows for authoring 3D viewing experiences that incorporate stylistic elements such as smooth animations between predefined viewing areas. Their approach gives the author a significant amount of control over what the end-user finally sees, while simultaneously providing the user with the feeling that they are actually navigating the scene, rather than simply playing a movie. Our present work shares an aspect of this previous body of research in that the system assists the author and viewer in the final viewing experience at the expense of ceding some, but not all, control to the system. However, unlike our current work, none of this prior art attempts to interactively utilize the information from previous visualizations to improve future visualizations.

2.2 Non-linear Projections

Non-linear projections have been applied to computer generated imagery for a variety of purposes such as image warping, 3D projections and multi-perspective panoramas. A survey of these applications is presented by Singh [29]. Of particular importance to this paper are interactive techniques for non-linear projection that assist in the visualization of complex 3D scenes.

Image warping [28, 33] approaches are inherently 2D, limiting their ability to explore a 3D scene from different viewpoints. Research in the area of nonlinear magnification for the purpose of visualization is well documented by Carpendale [4, 5].

Multi-perspective panoramas [23, 26, 34], are approaches for the construction of panoramas from camera motion. Panoramas are an effective way of visualizing landscapes with a wide angle of view, or for unfolding the detail of an object. These approaches are catered to capturing imagery using real cameras. However, they are, unfortunately, not well suited to interactive manipulation.

As an alternative, 3D deformations [12, 27] are widely used for manipulating 3D geometry. Interactive object deformation, like most modeling and animation tasks, usually requires camera manipulation as a secondary operation. View dependent distortions to scene geometry for animation and illustration have also been explored [16, 20, 22, 25]. Agrawala et al. [1] use multiple linear perspectives to define a composite rendering of a scene. In their approach each object is rendered in perfect linear perspective, possibly different from the perspective of other scene objects. While this leads to interesting artistic renderings it does not help and possibly impairs the visualization and understanding of a 3D scene.

A more recent approach to non-linear projection [29] combines a number of exploratory linear perspective cameras into a single interactive non-linear projection of a scene. We extend this work in Section 4.1 in our description of the Mosaic camera.

2.3 Interactive Data Mining

Data mining in its most general form is concerned with searching for interesting patterns within large datasets. Very generally, building a data mining system requires that two problems be solved: 1) appropriate and efficient storage of the relevant data to facilitate fast searching, and 2) the ability to extract interesting and relevant patterns from the stored data. In our current work, we are not concerned with innovating on either storage or pattern discovery data mining algorithms; rather we seek to apply established data mining techniques to a new problem domain. While data mining techniques have been applied to a variety of domains related to computer graphics, such as texture segmentation and computer vision, to the best of our knowledge only a few examples exist of its application to camera navigation [13]. With regards to storage of spatial data, the text by Laurini [19] provides a general overview. Shapiro and Frawley [24] discusses interesting examples of various issues in pattern discovery, where the patterns of interest are not known apriori. Of particular interest to us are various clustering algorithms [18], which can be broadly categorized into hierarchical or partitioning approaches. While hierarchical methods have been applied successfully in biological applications [18], one limitation is that they cannot undo previously determined clusters, making them less suitable for interactive applications. In contrast, partitioning methods, such as variants of k-means or k-mediod, try to find the *best* k partitions of the dataset that satisfy a given similarity criteria. We use a standard k-means algorithm in our work.

3. DATA MINING INFRASTRUCTURE

Before designing an infrastructure for data mining camera manipulation it is important to understand the parameters involved in camera manipulation and their impact on viewing a 3D scene. Conventional linear perspective cameras used to navigate through 3D scenes have a large number of parameters that can vary over time. Typically, camera manipulation involves control of the position, orientation, and focal length or center of interest of the camera. Additional parameters like clipping planes, aspect ratio, and other application specific data related to the camera may also be manipulated. The most comprehensive form of data collection would maintain a record of all parameters for all cameras at every instant in time. This approach would result in an intractable amount of data for complex applications and does not scale well over time. The data must, therefore, be pruned and structured for it to be usable within complex scenes over long periods of time, retaining the ability to be processed at interactive rates. We now consider techniques for collecting data that are both memory efficient and usable for interactive data mining.

3.1 Camera vs. Object Centric

Objects in complex 3D scenes also have parameters that can vary in time and these objects are often the subject of a camera's interest. In terms of the collection and structuring of camera manipulation data, one can thus look at the relationship between cameras and objects in two ways: 1) the objects a camera is looking at, or 2) the cameras with which an object or parts of an object is being looked at. As we show using examples in Section 4, some applications are viewer centric while others object centric. The focus of the application has a strong influence on the information mined from the collected data and thus affects the data structures used.

3.2 Spatial vs. Temporal Manipulation

The intent of the user when varying these parameters is sometimes explicitly clear and sometimes ambiguous, depending on the action performed by the user and the application context within which that action is performed. As an example, manipulating the camera in an interactive sculpting application is usually done to examine an operation just performed or to focus on a region about to be worked on. Here the path taken to the region of interest is of less importance than the final spatial configuration of the camera. In contrast, when generally evaluating a 3D scene, the pattern and paths along which the viewer navigates through the scene and its pacing in time, is as important as the regions over which a viewer dwells for periods of time.

3.3 Spatial Data Mining

There are a number of scenarios where it is desirable to know how the various parameters of a camera are distributed over time, disregarding their relative chronology. In such cases a simple scalable solution is to employ a spatial subdivision scheme over the domain of any varying camera parameter to aggregate the amount of time over which the parameter has a specific value. In practice we find that almost without exception the camera's position and orientation are the parameters typically manipulated by viewers. We thus embed our scene in a 6D subdivision of position and orientation. In Figure 4, the camera's position in space is captured by a spherical coordinate grid, which is well suited to sampling camera space within a range of distances between the camera and the object. The orientation of the viewing axis of the camera is captured by the subdivision of the two angles in the camera's local spherical coordinates, and camera tilt by an angle around the view axis.

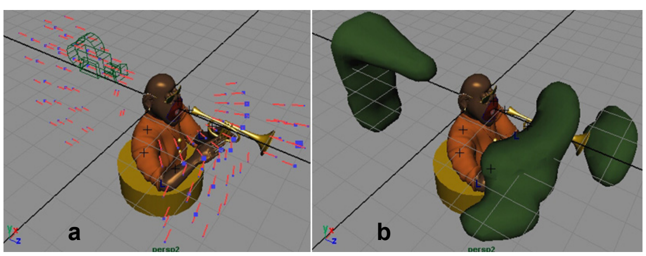

Figure 4. a) Tracking position, gaze, and focus of cameras over time. The size of the blue dots is proportional to time spent at that location. The pink line segments are the principal gaze directions over time. b) The green blobs are temporal iso-surfaces from the camera's position field.

As shown in Figure 3b, the center of interest of the camera is captured by a rectilinear grid, which embeds the scene being viewed. As the camera navigates through the scene it accumulates a value of time spent at each proximal grid locations using linear interpolation in each data dimension. In Figure 4a, camera position subdivision grid locations with non-zero accumulated time are shown with blue dots: the size of the dot proportional to the amount of time spent. The pink direction vector is a normalized, accumulated time weighted average of the vectors that subdivide the camera orientation at each position grid point. The pink lines thus indicate the principle gaze direction for every camera position grid point. The example in Figure 4 is typical for situations where we may wish to analyze camera manipulation behavior over long periods of time for one or more viewers. In interactive applications where data mining assists in performing the current task, older camera manipulations and ones less frequently used have less importance. We model this in our infrastructure as a simple decay parameter that attenuates the accumulated time values in the spatial data structure at every time step. This reduces the contribution of any given camera configuration as it gets older in time.

Given the collected data we can perform a number of interactive operations on it, the results of which can be used for a number of existing visualization techniques and some new ones that we describe in Section 4.

3.3.1 Clustering

The first most obvious data mining operation is to cluster the data to find camera configurations of local maxima, where viewers spend the majority of time.

Figure 5c shows the result of a k-means clustering algorithm [18] run on the dataset. The algorithm generates four clusters, the center of each cluster represented by the apex of a red cone, pointing at the principal gaze direction for the cluster. The green sphere around each cluster shows its spread (clusters of larger radius indicate a more even distribution of time over the camera configurations in the cluster). We can extract a larger number of more focused views by limiting the spread size of the clusters. Figures 5d-g show "through the lens" views of the mean camera configuration for the four clusters. These views correspond to the collected camera navigation data. The clustering algorithm is incremental and can be used interactively with ongoing data collection, while a camera is being manipulated.

There are large numbers of existing visualization techniques that can benefit from spatial data clustering. For example, some 3D graphics systems, such as *Alias' Maya*, allow viewers to bookmark camera views that they can switch to at will. Users often find that setting up bookmarks manually is cumbersome since it distracts them from their primary task. Incremental clustering allows us to automatically bookmark frequently used camera views unobtrusively, while the user manipulates the camera as part of their primary task. Attaching a decay parameter to the data collection allows old and rarely used bookmarks to be purged automatically, increasing the usability of the bookmarking functionality in current 3D systems. Clusters can also be used to create camera attractor fields as described by Hanson and Wernet [13, 32], where both the center of the field and its region of influence can be controlled by the cluster. Finally, in Section 4.1, we demonstrate the use of clusters to construct non-linear mosaic projections of 3D scenes.

3.3.2 Isosurfaces

Figure 4b shows an isosurface representing camera positions in space with equal accumulated time. While clusters are used to find local maxima or minima in the collected data, isosurfaces allow us to extract paths and regions of uniform camera manipulation. Figure 6a shows an isosurface corresponding to camera positions with a small amount of accumulated time.

Figure 6b shows an isosurface with a greater accumulated time on the same dataset. We can now extract isocurves on the isosurfaces based on the camera orientation. Curves of constant principal camera orientation that are perpendicular to the curve can indicate a camera pan through the scene. Curves with a principal camera orientation tangent to the curve are typically generated by dollying the camera. Figure 6c shows a more complex isocurve with the property that the camera orientation along the curve is directed at a given point. As seen in Figure 6c the curve helps us reconstruct the path representing a tumble of the camera about the origin. StyleCam [3] used a similar approach where cluster centers defined “money shots”. Isosurfaces clipped at the cluster spread boundary can be used to define the surfaces onto which the cluster center is projected. Finally, isocurves can be used to define transition paths between camera surfaces that share a common isosurface.

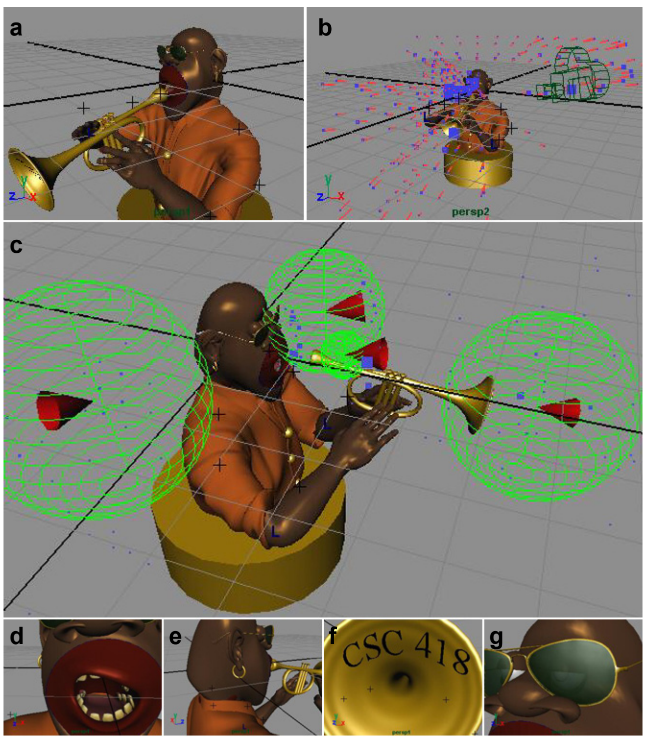

Figure 5. a) View through the camera navigating through the scene in 5b. c) Results of data clustering. The cones point in the direction of the average gaze direction for the cluster with their apex at the center. The green spheres show the radius of the cluster. d-g) Through the lens views for the camera clusters.

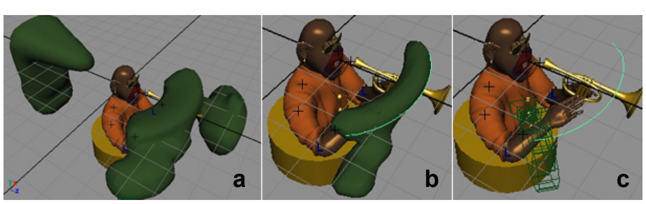

Figure 6. a) Volume envelope for little time spent at a location. b) More focused isosurface of higher time value, with extracted isocurve that is a reconstruction of a camera path shown in 6c.

3.4 Temporal Data Mining

There are also applications where we wish to track the chronology of manipulations, as we will see with the sticky-camera example in Section 4.2. In such cases we use a simple event list of camera configurations over time. For most interactive data mining applications related to camera navigation, however, it is both sufficient and necessary to constrain the event list to a finite sized moving window of time. Events older than the window either decay in importance or get aggregated into spatial data structures as described in Section 3.3.

3.5 Data Filtering

The various data mining operations are fairly sensitive to noise and errant camera motion. Unfortunately, the interactive and exploratory nature of camera manipulation make it next to impossible to collect data that is devoid of jerky motion, unintentional camera moves and other camera noise. We typically need to apply data filters to improve the quality of the data to be able to apply various data mining algorithms successfully. Most standard data filtering algorithms used in image processing can be applied to our datasets, given that we are using regular subdivision grids. As an example, Figure 7a shows an example of jerky camera movements that result in the disjoint camera movement isosurfaces. Figure 7b shows the results of the isosurfaces after running a Laplacian smoothing filter on the dataset to even out the non-uniformity in the data.

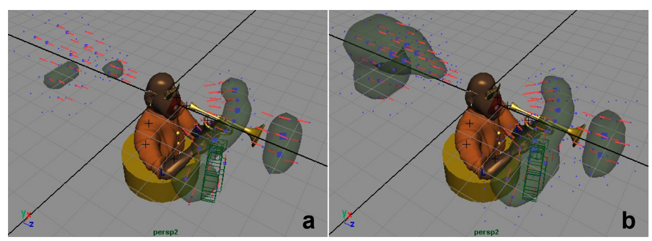

Figure 7. a) Incomplete data sampling from jerky camera motion results in poor reconstruction. b) Reconstruction after a Laplacian filter has been applied to the data retrieved in 7a.

4. VISUALIZATION APPLICATIONS

Section 3 presented a variety of data structuring and mining techniques that are well suited to interactive camera manipulation applications, as seen by their use with existing visualization techniques. We now use this infrastructure to describe three novel techniques for the interactive visualization of complex scenes, that employ elements of non-linear projection.

4.1 Mosaic Camera

The Mosaic camera is inspired by the fact that while linear perspective is an excellent visualization tool providing consistently understandable depth cues for parts of a 3D scene, it is restrictive for the overall visualization of complex shapes. Singh [29] presented an approach to building non-linear projections of scenes as compositions of multiple exploratory linear perspectives. Each exploratory camera is defined using parameters such as position, orientation and center of interest. A viewport for each camera maps its perspective view onto a common projection canvas. For every point in the scene, a weight value is first computed for each exploratory camera based on user-controlled functions such as proximity to camera's center of interest. The weights for all cameras were then normalized and a virtual camera constructed as a weighted interpolation of the parameters defining the exploratory cameras. The point was then projected onto a common canvas using the virtual camera and a similarly weight interpolated viewport transformation. While this is a general approach to non-linear projection the number of variables that may be controlled to construct a non-linear visualization is quite daunting: we must first pick the number and settings for the different exploratory cameras. Then we must define viewports for them, the size and layout of which is implicitly constrained by the position and orientation of the exploratory cameras relative to each other. Finally we must figure out the relative weighting or importance of the cameras and weight computation functions for each camera.

Here we use the spatial subdivision scheme presented in Section 3.3 to conceptually capture the non-linear projection model described above. Let each camera position grid point represent the position of an exploratory camera with its orientation represented as the principal orientations as shown in Figure 4. The viewport for each camera is represented by a spherical projection of the square around it as shown in Figure 8. Points in the scene are weighted by their distance from the line segment between the camera's position and its center of interest. This weight function is attenuated by an overall camera weight that is proportional to the time accumulated at the camera's grid position. Any point P is thus projected to a weighted summation of the projection of the point through every grid point camera.

While this formulation is conceptually robust, it is not efficient as there maybe a huge number of exploratory cameras to consider for every point in the scene. Instead we use clusters to define the exploratory views and use the cluster spread to influence both the falloff function for a point as well as the viewport size. Figure 1 shows the results of such a formulation using 3 cluster cameras.

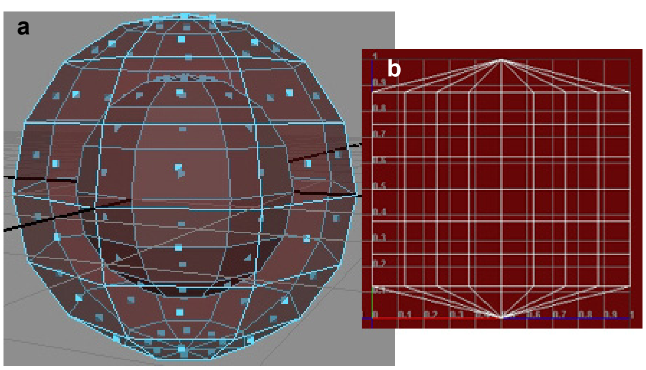

Figure 8. a) Blue dots represent exploratory cameras in a spherical grid. The polygon around it represents the viewport transformation laid out as a spherical projection as in 8b.

4.2 Sticky Cmera

In many applications, camera manipulation is used to bring parts of a scene outside the current viewing frustum of the camera into view. This is either done at the expense of parts currently in view by tumbling or panning the camera, or by an overall loss of view detail by dollying or zooming the camera. The sticky camera attempts to bring new parts of the scene into view after the camera is manipulated, while preserving the view properties of regions in focus prior to the camera manipulation. This is accomplished by non-linear projection.

Let C_b, M_b and V_b represent the eye-space, projection and viewport transformation matrices of the camera before manipulation and C_a, M_a and V_a the corresponding matrices after manipulation. For a point P to appear in the camera view after manipulation as it did in the camera view before manipulation it would need to be deformed to a point $P' = P\ (C_b\ M_b\ V_b\ )(C_a\ M_a\ V_a\ )^{-1}$. The aim of the sticky camera, is to preserve the view perspective of points prior to manipulation but these points must moved on-screen somewhat to create space for new parts of the scene that are coming into view. With this in mind we add a relative viewport translation V_r proportional to the $-T$, where T is the displacement of the camera's center of interest as a result of camera manipulation, i.e. $P' = P\ (C_b\ M_b\ V_b\ )V_r(C_a\ M_a\ V_a\ )^{-1}$. Further all points in the scene have a weight w_P that is based on camera parameters, for example, a falloff function of the distance of the point from the camera's center of interest. For sticky-camera behavior thus any point is transformed to a point $P_{def} = P + w_P(P'-P)$. Such a non-linear projection formulation works well for a single camera manipulation step. Camera manipulation takes place as a stream of camera events and we need to employ data structures for temporal mining as in section 3.4 so as not to have to store an increasingly long array of camera configurations. In this case we use a simple temporal window size of two. Deformation and weight values are aggregated in an object centric fashion on a per point basis as D_P and an overall weight value, o_P. D_P += $o_P P'$ and o_P += w_P. This aggregated deformation is added into the deformation for a later step as $P_{def} = P + w_P(P'-P) + D_P\ (V'_r(C_b\ M_b\ V_b\ )^{-1}\ )^{-1}(V_r(C_a\ M_a\ V_a\ )^{-1}) - o_P P$, where V'_r is the relative viewport translation for the last timestep (C_b,M_b,V_b are C_a,M_a,V_a from last timestep). This allows us to aggregate deformation data in an object centric way with just a single vector D and scalar weight o, per point.

Results of the sticky-camera can be seen in Figure 2 and the animation still in Figure 9.

4.3 Fisheye Camera

The fisheye-camera is motivated by the observation that part of an object viewed for a longer period of time should be presented in greater relative detail on screen. The concept is illustrated first on a 2D grid and the concept extends trivially into three and higher dimensions. Suppose a camera (2D viewport in this case) viewing parts of the (m+1)x(n+1) grid of unit cell dimensions has accumulated time values t_{ij} at grid location (i,j). We deform the grid location (i,j) to $(i*(h_{ij}/h_{mj}), j*(v_{ij}/v_{in}))$, where h_{ij} is the sum of accumulated times along the j*th* row from h_{0j} to h_{ij} and v_{ij} is the sum of accumulated times from along the i*th* column v_{i0} to v_{ij}. The grid is initialized with $t_{ij}=1$ for all (i,j).

The deformed grid has non-uniform sizing of grid cells to reflect the amount of time spent looking at any grid point. In 3D we subdivide scene space with a rectilinear grid structure and use a radial falloff function around the camera's center of interest to compute a value of focus-time for each grid point. We then use the grid structure to define a free-form deformation lattice [27], resulting in a smooth deformation of all objects in the scene to reflect the time spent viewing different parts of it (see Figure 3).

5. DISCUSSION and CONCLUSIONS

The concepts in this paper have been implemented as plug-ins to the animation system *Maya4.5*. The use of a commercial animation system makes these visualization ideas instantly usable within a current user workflow. The mosaic camera and sticky camera are currently being used in experiments within a commercial animation production entitled "Ryan". Figure 9 shows a variant of the sticky camera in a bathroom scene, where the horizontal fringes of the view are sticky, causing a non-linear distortion to the otherwise linear perspective camera view.

For the mosaic and sticky camera, there is an implicit connection between the spatial relationships of cameras in 3D and their relative viewport layout in 2D. It is thus possible to construct scenarios in our current formulation, where parts of a scene being viewed by a sticky camera *catch-up* and penetrate a region that was being viewed previously. In practice we pick conservative viewport movements commensurate with the motion of the camera in 3D to avoid scene projections from self-penetrating. The fish-eye camera on the other hand is mathematically formulated as a non-overlapping deformation of space. A similar theoretically robust solution for sticky and mosaic camera is subject to future work.

Effective visualization of complex 3D scenes is a difficult task. In this paper we look to automated approaches to simplify both camera navigation and 3D scene visualization, by data mining past and ongoing camera manipulation. We put particular emphasis to interactivity and as such all the techniques proposed, from our data collection and mining infrastructure, to visualization examples, can be run at interactive rates for complex scenes. The examples in the paper illustrate the significant potential of data mining camera navigations as well as the use of non-linear projection for the presentation of complex 3D scenes. Controlling non-linear projections can be difficult and this paper presents a number of examples where the integration of spatial and temporal aspects of linear perspective camera navigation automatically provides us with understandable non-linear projections of scenes.

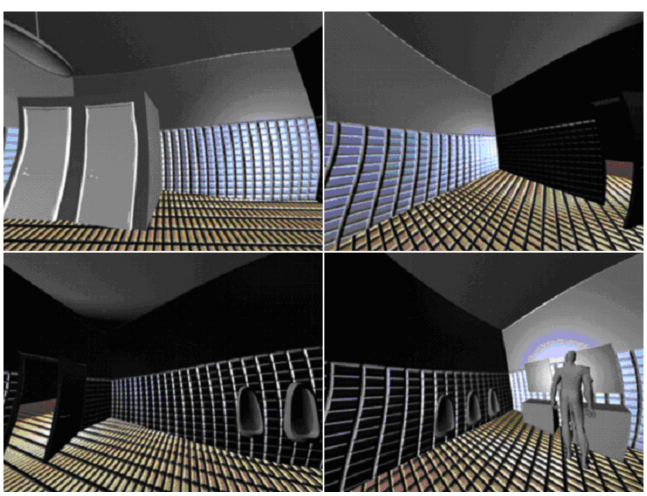

Figure 9. Sticky camera scene from animation "Ryan"

6. REFERENCES

[1] Agrawala, M., Zorin, D., & Munzner, T. (2000). Artistic multiprojection rendering. Eurographics Rendering Workshop. p. 125-136.

[2] Bowman, D.A., Johnson, D.B., & Hodges, L.F. (1997). Travel in immersive virtual environments. IEEE VRAIS Virtual Reality Annual International Symposium. p. 45-52.

[3] Burtnyk, N., Khan, A., Fitzmaurice, G., Balakrishnan, R., & Kurtenbach, G. (2002). StyleCam: Interactive stylized 3D navigation using integrated spatial and temporal controls. ACM UIST Symposium on User Interface Software and Technology. p. 101-110.

[4] Carpendale, M.S.T., & Montagnese, C.A. (2001). A framework for unifying presentation space. ACM UIST Symposium on User Interface Software and Technology. p. 61-70.

[5] Carpendale, S. (1999). A framework for elastic presentation space. Ph.D. Dissertation thesis. Simon Fraser University.

[6] Darken, R., & Sibert, J. (1996). Wayfinding strategies and behaviours in large virtual worlds. ACM CHI Conference on Human Factors in Computing Systems. p. 142-149.

[7] Drucker, S.M., & Zeltzer, D. (1994). Intelligent camera control in a virtual environment. Graphics Interface. p. 190-199.

[8] Elvins, T., Nadeau, D., Schul, R., & Kirsh, D. (1998). Worldlets: 3D thumbnails for 3D browsing. ACM CHI Conference on Human Factors in Computing Systems. p. 163-170.

[9] Furnas, G. (1986). Generalized fisheye views. ACM CHI Conference on Human Factors in Computing Systems. p. 16-23.

[10] Galyean, T.A. (1995). Guided navigation of virtual environments. ACM I3D Symposium on Interactive 3D Graphics. p. 103-104.

[11] Gliecher, M., & Witkin, A. (1992). Through-the-lens camera control. ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques. p. 331-340.

[12] Gomez, J., Darca, L., Costa, B., & Velho, L. (1998). Warping and morphing of graphical objects. Morgan Kaufmann. San Francisco.

[13] Hanson, A.J., & Wernet, E. (1997). Constrained 3D navigation with 2D controllers. IEEE Visualization. p. 175-182.

[14] He, L., Cohen, M.F., & Salesin, D. (1996). The virtual cinematographer: a paradigm for automatic real-time camera control and directing. ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques. p. 217-224.

[15] Igarashi, T., Kadobayashi, R., Mase, K., & Tanaka, H. (1998). Path drawing for 3D walkthrough. ACM UIST Symposium on User Interface Software and Technology. p. 173-174.

[16] Inakage, M. (1991). Non-linear perspective projections. IFIP WG5.10. p. 203-215.

[17] Jul, S., & Furnas, G. (1998). Critical zones in desert fog: aids to multiscale navigation. ACM UIST Symposium on User Interface Software and Technology. p. 97-106.

[18] Kaufman, L., & Rousseeuw, P. (1990). Finding groups in data: An introduction to cluster analysis. John Wiley & Sons.

[19] Laurini, R., & Thompson, D. (1992). Fundamentals of spatial information systems. Academic Press. .

[20] Levene, J. (1991). A framework for non-realistic projections. Masters Thesis thesis. MIT.

[21] Mackinlay, J., Card, S., & Robertson, G. (1990). Rapid controlled movement through a virtual 3D workspace. ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques. p. 171-176.

[22] Martin, D., Garcia, S., & Torres, J. (2000). Observer dependent deformations in illustration. Non-Photorealistic Animation and Rendering (NPAR). p. 75-82.

[23] Peleg, S., Rousso, B., Rav-Acha, A., & Zomet, A. (2000). Mosaicing on Adaptive Manifolds. IEEE Transactions on Pattern Analysis and Machine Intelligence. 22(10). p. 1144-1154.

[24] Piatetsky-Shapiro, G., & Frawley, W. (1991). Knowledge discover in databases. AAAI/MIT Press. .

[25] Rademacher, P. (1999). View-dependent geometry. SIGGRAPH Conference on Computer Graphics and Interactive Techniques. p. 439-446.

[26] Rademacher, P., & Bishop., G. (1998). Multiple-center-of-projection images. SIGGRAPH Conference on Computer Graphics and Interactive Techniques. p. 199-206.

[27] Sederberg, T., & Parry, S. (1986). Free-form deformation of solid geometric models. ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques. p. 151-160.

[28] Seitz, S., & Dyer, C. (1996). View morphing: Synthesizing 3D metamorphoses using image transforms. SIGGRAPH Conference on Computer Graphics and Interactive Techniques. p. 21-30.

[29] Singh, K. (2002). A fresh perspective. Graphics Interface. p. 17-24.

[30] Steed, A. (1997). Efficient navigation around complex virtual environments. ACM VRSTConference on Virtual Reality Software and Technology. p. 173-180.

[31] Tan, D., Robertson, G., & Czerwinski, M. (2001). Exploring 3D navigation: combining speed-coupled flying with orbiting. ACM CHI Conference on Human Factors in Computing Systems. p. 418-425.

[32] Wernert, E.A., & Hanson, A.J. (1999). A framework for assisted exploration with collaboration. IEEE Visualization. p. 241-248.

[33] Wolberg, G. (1992). Digital image warping. IEEE Computer Society Press. .

[34] Wood, D., Finkelstein, A., Hughes, J., Thayer, C., & Salesin, D. (1997). Multiperspective panoramas for cel animation. ACM SIGGRAPH Conference on Computer Graphics and Interactive Techniques. p. 243-250.

[35] Zeleznik, R., & Forsberg, A. (1999). UniCam - 2D Gestural Camera Controls for 3D Environments. ACM I3D Symposium on Interactive 3D Graphics. p. 169-173.

[36] Zeleznik, R., Forsberg, A., & Strauss, P. (1997). Two pointer input for 3D interaction. ACM I3D Symposium on Interactive 3D Graphics. p. 115-120.