Annotating Gigapixel Images

Qing Luan [*] USTC [†] Hefei, China qing_182@hotmail.com

Steven M. Drucker Microsoft Live Labs Redmond, WA sdrucker@microsoft.com

Johannes Kopf University of Konstanz Konstanz, Germany kopf@inf.uni-konstanz.de

Ying-Qing Xu Microsoft Research Asia Beijing, China yqxu@microsoft.com

Michael F. Cohen Microsoft Research Redmond, WA mcohen@microsoft.com

ABSTRACT

Panning and zooming interfaces for exploring very large images containing billions of pixels (*gigapixel* images) have recently appeared on the internet. This paper addresses issues that arise when creating and rendering auditory and textual annotations for such images. In particular, we define a distance metric between each annotation and any view resulting from panning and zooming on the image. The distance then informs the rendering of audio annotations and text labels. We demonstrate the annotation system on a number of panoramic images.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

General terms: Design, Human Factors

Introduction

Images can entertain us and inform us. Enhancing images with visual and audio annotations can provide added value to the images. Supporting the authoring of and delivering of annotations in online images has taken many forms ranging from advertising, to community annotations of images in Flickr, to the multiple types of annotations embedded in applications depicting satellite imagery such as Google Earth and Virtual Earth. Last year we saw the introduction of systems to create and view very large (gigapixel) images [10], along with the introduction of new viewers for such images (e.g., Zoomify and HD View). Much like in the earth browsers, when viewing gigapixel imagery, only a tiny fraction of the image data is viewable at any one time. For example, when viewing a 5 gigapixel image on a 1 megapixel screen only $1/5000^{th}$ of the data is ever seen. Exploring the imagery is supported by a panning and zooming interface.

In this short paper, we explore many of the issues that arise when annotating and rendering annotations within very large images. Our main contribution is a model to represent the viewer's *perceptual distance* from the objects being annotated in the scene while continuously panning and zooming. This model informs an annotation rendering system for when and how to render both auditory and visual annotations. We demonstrate the annotation system within HD View[1], an internet browser hosted viewer based on the ideas in [10].

There are many things this paper does not address including automating the addition of annotations from web-based content, searching annotations within images, or the automated label layout problem. Each of these problems have been addressed in other contexts and the lessons learned from them can be applied here. Instead we focus on what is unique about annotations within the continuous panning and zooming environment. Thus, in addition to annotations of gigapixel imagery, our work is perhaps most directly applicable to the earth browsing systems.

Related Work

Our work draws from a number of areas ranging from Zoomable UIs, to map labeling, human psychophysics, and virtual and augmented reality systems.

Zoomable UIs (ZUIs) have been popularized by such systems as PAD [12] and PAD++ [2] which introduce the idea of navigating through a potentially infinite 2D space by panning and zooming a small view into that space. They include hiding objects when they are below a certain minimal magnification threshold. In most ZUIs, the threshold is based solely on the screen extent that an object occupies as opposed to some notion of perceptual distance to an object as we model.

There has been a wide body of literature focusing on label placement on maps beginning as early as 1962 with Imhof [8] and continuing today [18]. Much of this work is concerned with the automatic placement of labels which minimize overlap while optimizing nearness to named features. A thorough bibliography can be see at the Map-Labeling Bibliography[2]. We generally do not address the issue of laying out large numbers of labels as our model implicitly controls the density of labels before layout.

Interactive map systems such as Google Earth and Virtual Earth are perhaps the closest to our work. They provide a panning and zooming interface in which various kinds of annotations appear and disappear. Work in this area has focused on avoiding visual artifacts such as label popping as well as assuring that labels will appear at interactive rates [3]. Some work has been devoted to avoiding visual clutter which can impair map reading performance [13].

There has been extensive work on adding audio and textual labels to both virtual and augmented reality systems [7, 1, 4, 19]. It is clear that a greater sense of presence can be achieved by adding binaural sound rendering to a virtual environment [11] and there have been systems that focus on the efficient rendering of spatialized audio in complicated virtual environments [17]. Much of this work can be guided by the

[*]This work was done while Qing Luan was a visiting student at Microsoft Research and Microsoft Research Asia

[†]USTC: Unversity of Science and Technology of China

UIST'08, October 19–22, 2008, Monterey, California, USA..

[1]http://research.microsoft.com/ivm/HDView.htm

[2]http://www.math.drofnats.edu/riemann.ps

Figure 1: A Seattle panorama and associated very coarse hand painted depth map.

psychophysics of human sound localization [5]. Likewise, the automatic placement of labels within both virtual reality and augmented reality systems needs to consider such issues as frame to frame coherence, label readability, appropriate view selection, and occlusion of objects and other labels in the environment. While our work is related, we emphasize instead rapid annotation of large images instead of annotation of a true 3D environment and thus cannot use more straightforward physical models for audio rendering or label placement.

Annotations and Views

Due to their size, gigapixel images are typically never stored as a single image, but rather are represented as a multi-resolution pyramid of small tiles that are downloaded and assembled on-the-fly by the viewer. We use HD View that runs within multiple internet browsers. Conceptually, the image consists of a grid of pixels in $x \times y$ where, depending on the projection (perspective, cylindrical, spherical) of the underlying image, there is a mapping from x and y to directions in space. Given a virtual camera, HD View renders the appropriate view depending on the specific orientation, pan and zoom parameters of the virtual camera. For the purposes of the discussion here, we normalize x to lie in $[0, 1]$ while y varies from 0 to y_{max} depending on the aspect ratio of the image. Optionally, depth values across the image, $d(x, y)$, can be provided. In the annotation system, d represents the log of the scene depth. In our examples, a rough depth map consisting of three to five (log) depth layers is painted by hand and low-pass filtered (see Figure 1). The system is quite forgiving of inaccuracies.

Gigapixel Annotations

Annotations in gigapixel images reference objects within an image. For example, in a cityscape, an annotation may refer to a region of the city, a building, or a single person on the street that cannot be seen due to its small size when fully zoomed out. Thus, just as a view has a position and an extent defined by the zoom level, so does an annotation. The annotations themselves are specified from within the interactive viewer while panning and zooming. The user simply draws a rectangle in the current view indicating the extent of the object being annotated. The annotation position, (x_A, y_A) is set as the center of the rectangle. The annotation's "field of view", f_A is set by the size of the annotation rectangle,

$$f_A = \sqrt{(x_{right} - x_{left}) \cdot (y_{top} - y_{bottom})}$$

of the annotation rectangle. Thus an annotation can be said to be located at (x_A, y_A, f_A, d_A) where $d_A = d(x_A, y_A)$ (See Figure 2).

Currently, annotations can be one of three types: a text label, an audio loop, or a narrative audio. Many other types can be supported within the same framework such as hyperlinks, links to images, etc. For audio, the annotations are associated with a .wav file. Text label annotations contain a text string as well as an offset within the rectangle and possible leader line to guide final rendering.

Figure 2: Parameters of an annotation and view.

Gigapixel Image View

Given the pan and zoom, the center of the view has some coordinate (x_v, y_v) and some field of view f_v relative to the full image that defines the x and y extents of the view. We say that $f_v = 1$ when the image is fully zoomed out and visible, and $f_v = 0.5$ when zoomed in so half the width of the full image is within the browser frame, etc. Thus, at any zoom level $f_v = x_{right} - x_{left}$ of the current view.

Depth of the Viewer We set the depth of the viewer to be the value of the depth map at the center of the screen, $d_v = d(x_v, y_v)$. As we discuss later, this depth value plays an increasing role as we zoom in to the image. The viewpoint is in reality fixed, but perceptually as one zooms into the image, there is also a perception of moving closer to the objects in the scene. The word for this perceived motion is *vection*: the perception of self-motion induced by visual stimuli [6]. Vection has been studied primarily for images in which the camera actually moves forward inducing motion parallax between near and far objects. In our case, zooming induces outward optical flow but no parallax. The ambiguity between narrowing the field of view (zooming) and dollying into the scene (moving forward) results in similar vection effects. This is supported by a number of studies [15, 14, 16]. A single center value to represent the depth was chosen to avoid on-the-fly "scene analysis" to keep the viewing experience interactive.

Perceived Field of View We make one further modification to the specification of the view. As the user rapidly pans and zooms, it is hypothesized that users are more aware of larger objects and when stopped on a particular view they become more aware of smaller objects. This notion is supported in the perception literature by studies on the changes in the *spatial contrast sensitivity function* (SCSF: our ability to distinguish the fluctuations fine sine gratings) between moving and still images [9]. We cannot see so much high frequencies in an image when it is moving across our visual field as when it is still. Conversely, we become more aware of very low frequencies when the image moves.

Both of these perceptual effects are captured by establishing a *perceived field of view value*, $\tilde{f}_v$, that grows with motion and shrinks when still. This is implemented as follows. A field of view multiplier, m_f at time zero is initialized to be 1.0, $m_f(0) = 1$. At each time step, this multiplier is increased if the view is changing and decreased if the view is static.

More formally, a variable $m(t)$ is an indicator of motion. $m(t) = c_f$ if there has been any panning or zooming motion of the view between time $t - 1$ and time t, and $m(t) = 1/c_f$

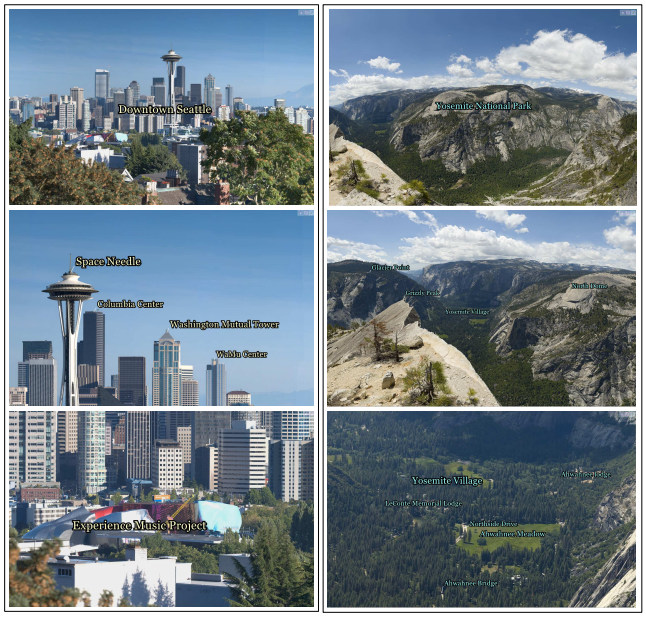

Figure 3: Three views of an annotated gigapixel image of Seattle (a) and Yosemite (b)

if the view is still. c_f is a parameter that controls the strength of the maximum and minimum values the multiplier converges to. We have set c_f to 1.5 which corresponds roughly to changes in the SCSF for motion of 2 degrees of visual angle per second. Further study could be done to see if this value should vary based on motion speed. Thus, at each time step:

$$m_f(t) = \beta\, m(t) \;+\; (1-\beta)\, m_f(t-1)$$

and finally: $\tilde{f}_v = m_f\, f_v$

where β controls how fast the motion effect varies. A value of β approximately one over the frame rate works well, or approximately 0.3. Thus, as m_f varies slowly between c_f and $1/c_f$, the effective zoom grows and shrinks accordingly. Thus a view is fully specified by its position, perceptual size, and the depth value at its center. This is captured by the tuple $(x_v, y_v, \tilde{f}_v, d_v)$ (See the blue doted rectangle in Figure 2).

Annotation Strength

Given a gigapixel image, a set of annotations, the current view, and some view history, the annotation rendering system decides which annotations to render (visual or audio), what *strength* each should have, and where to place the annotation (label position or spatialized stereo). The strength of each annotation is inversely correlated to the *distance* between the current view and the annotation.

Distance Between an Annotation and a View

To determine the strength with which to render each annotation, we begin by computing four distance values between the view and the annotation:

$Xdist = |x_A - x_v|$ describes the horizontal offset between the view and each annotation.

$Ydist = |y_A - y_v|$ describes the vertical offset between the view and each annotation.

$Fdist = |\tilde{f}_v - f_A|/\tilde{f}_v$ if $\tilde{f}_v > f_A$ (while zooming in to the field of view of the annotation), and $Fdist = |\tilde{f}_v - f_A|/(1-\tilde{f}_v)$ otherwise (i.e., when we are zooming in beyond the field of view of the annotation). Intuitively, $Fdist$ measures how large the object being annotated is relative to the view, becoming zero when the object would fill the screen.

$Ddist = c_d|d_A - d_v| \cdot (1-\tilde{f}_v)$, thus as we zoom in, (i.e., $\tilde{f}_v$ gets smaller), the differences in depths takes on an increasing role. A narrow field of view invokes a stronger sensation of being at the depth of the object than we have with a wider angle view. c_d normalizes the depth difference term, typically set to $1/(d_{max} - d_{min})$.

$$D = \sqrt{Xdist^2 + Ydist^2 + Fdist^2 + Ddist^2}$$

Initial Strength of an Annotation

Finally, the initial strength, A, of each annotation drops off with distance:

$$A = exp(-D/\sigma_D)$$

where σ_D controls the drop off of the annotations with distance. We have found a default value of $\sigma_D = 0.1$ to work well. σ_D, however, is the one parameter it makes sense to put in the user's hands. By varying from σ_D from small values to large, the user can control the whether only those annotations in the immediate central view (i.e., have small D values) carry any strength, or with larger σ_D, all annotations carry more even strengths.

Ambient Annotations

In addition to the standard annotations, there is one additional *ambient* audio and label annotation. These annotations are global and carry a constant weight, A_0, which we currently set to 0.2. The ambient audio annotation provides background audio. The ambient label annotation is typically just a *null* annotation. The ambient audio volume and influence of the null text annotation diminish as other annotations gain strength due to the normalization described next.

Normalization

To maintain an approximate constancy of annotations we normalize the strength of each annotation relative to the total of the strengths including the ambient term.

$$\bar{A}_i = A_i / \textstyle\sum_i A_i$$

This normalization is done separately for the set of audio annotations and the set of text label annotations.

Hysteresis

Finally, we add a hysteresis effect to the strengths associated with each annotation

$\tilde{A}(t) = \alpha_+\, \bar{A}(t) \;+\; (1-\alpha_+)\, \tilde{A}(t-1)$ for rising strengths,

$\tilde{A}(t) = \alpha_-\, \bar{A}(t) \;+\; (1-\alpha_-)\, \tilde{A}(t-1)$ for falling strengths,

so that the final strength of each annotation varies slowly. We set $\alpha_+ = 0.2$, and $\alpha_- = 0.05$. The final strength $\tilde{A}$ is guaranteed to lie in the interval $[0, 1]$.

Rendering the Annotations

Given the strength, $\tilde{A}$, for each annotation, we are now ready to render the annotations. The panorama is rendered by HD View using DirectX within an internet browser. Text labels are drawn in the overlay plane.

Audio Loop Annotations

Audio loop (ambient) annotations are rendered with volume directly correlated with the strength $\tilde{A}$. We do, however,

modulate the left and right channels to provide stereo directionality to the audio. Signed versions of $Xdist$ and $Ddist$

$Xdist_{signed} = x_A - x_v$
$Ddist_{signed} = \mathrm{Sign}(\mathrm{d_A} - \mathrm{d_v})(\mathrm{c_d}|\mathrm{d_A} - \mathrm{d_v}|)$

provide the angle $atan(Xdist_{signed}/Ddist_{signed})$ between the view direction and the annotation center which determines the relative left and right volumes.

Audio Narrative Annotations

Audio narrative annotations are intended to be played linearly from the start onward. We set two thresholds on the strength. One specifies when a narrative annotation should be triggered to start. When triggered, the narrative begins at full volume. At some lower strength threshold, the narrative begins to fade in volume over 3 seconds until it is inaudible. If the user moves back towards the narrative source while it is still playing the narrative continues and regains volume. Once it has stopped, however, the narrative will not begin again until some interval (currently set to 20 seconds) has passed. As in the looping audio annotations, the narrative is also modulated in stereo. Finally, if one narrative is playing, no other narrative can be triggered to play.

Text Labels

The appearance and disappearance of text labels are also triggered by thresholds. As in the narrative annotations text annotations are triggered to fade in over one second at a given strength value. They are triggered to fade over one second at a somewhat lower threshold.

There are two trivial ways to set the text size. It can have a fixed screen size, or can be a fixed size in the panorama coordinates. Unfortunately, in the former case, even though the true size does not change, it will appear to shrink as one zooms in since the context is growing around it. In the latter case, the text will be too small to read when zoomed out and will appear to grow and seem enormous when zoomed in. Instead, we compromise between these two cases. More specifically,

$$\mathrm{TextSize} = \mathrm{c_{text}}\,(\gamma + (1-\gamma)\mathrm{z_A}/\mathrm{z_v})$$

where our defaults are $c_{text} = 16\mathrm{point}$ and $\gamma = 0.5$. This results in a perceptually more uniform text size even though the text in fact grows as one zooms in. Although there has been some earlier work on determining label size based on object importance [19], we have not seen any notion of the dynamic perceptual size constancy during zooming applied before.

Parameter Setting

As all parameters in the system can be set by educated intuition, they required very little trial and error. The parameters had the same values in all examples. However, the ambient and hysteresis parameters are somewhat a matter of taste: smaller values lead to more responsive but jumpier behavior.

Results and Discussion

We have demonstrated a system for annotating very large images viewed within a panning and zooming interface. Figure 3 shows some screen shots of text annotations within large panoramas. For demos of the system, please visit our website [3].

[3] `http://research.microsoft.com/~cohen/GigapixelAnnotations/GigaAnnotations.htm`

Our primary contribution is a distance function between the individual annotations and views of the image that guide the rendering of both audio and text annotations.In this short tech note we do not report on a formal study. However, we demoed the application at a 2-day event to 5,000 attendees to great enthusiasm. A more formal study would certainly help confirm what we have observed.

REFERENCES

1. Ronald Azuma and Chris Furmanski. Evaluating label placement for augmented reality view management. In *Proc. IEEE and ACM Int'l Symp. on Mixed and Augmented Reality*, 2003.
2. Benjamin B. Bederson and James D. Hollan. Pad++: A zoomable graphical interface. In *Proceedings of ACM Human Factors in Computing Systems Conference Companion*.
3. Ken Been, Eli Daiches, and Chee Yap. Dynamic map labeling. *Transactions on Visualization and Computer Graphics*, 2006.
4. Blaine Bell, Steven Feiner, and Tobias Höllerer. View management for virtual and augmented reality. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, 2001.
5. J. Blauert. Spatial hearing: The psychophysics of human sound localization. 1983.
6. T. R. Carpenter-Smith, R. G. Futamura, and D. E. Parker. Inertial acceleration as a measure of linear vection: an alternative magnitude estimation. *Perception and Psychophysics*.
7. K. K. P. Chan and R. W. H. Lau. Distributed sound rendering for interactive virtual environments. In *International Conference on Multimedia and Expo*, 2004.
8. Eduard Imhof. Die anordnung der namen in der karte. *Internat. Yearbook of Cartography*, pages 93–129, 1962.
9. D. H. Kelly. Motion and vision. ii. stabilized spatio-temporal threshold surface. *J. Opt. Soc. Am.*, 1979.
10. Johannes Kopf, Matt Uyttendaele, Oliver Deussen, and Michael Cohen. Capturing and viewing gigapixel images. *ACM Transactions on Graphics*, 2007.
11. P. Larsson, D. Vastfjall, and M. Kleiner. Better presence and performance in virtual environments by improved binaural sound rendering. In *proceedings of the AES 22nd Intl. Conf. on virtual, synthetic and entertainment audio*, 2002.
12. Ken Perlin and David Fox. Pad: An alternative approach to the computer interface. *Computer Graphics*, 27.
13. R. J. Phillips and L. Noyes. An investigation of visual clutter in the topographic base of geological map. *Cartographic J.*, 19(2):122–131, 1982.
14. H. J. Sun and B. J. Frost. Computation of different optical velocities of looming objects. *Nature Neuroscience*, 1998.
15. Spratley J. Telford, L. and B. J. Frost. The role of kinetic depth cues in the production of linear vection in the central visual field. *Perception*, 1992.
16. L. C. Trutoiu, S.D. Marin, B. J. Mohler, and C. Fennema. Orthographic and perspective projection influences linear vection in large screen virtual environments. page 145. ACM Press, 2007.
17. N. Tsingos, E. Gallo, and G. Drettakis. Perceptual audio rendering of complex virtual environments. 2004.
18. A. Woodruff, J. Landay, and M. Stonebraker. Constant information density in zoomable interfaces. In *Proceedings of the Working Conference on Advanced Visual interfaces*, 1998.
19. Fan Zhang and Hanqiu Sun. Dynamic labeling management in virtual and augmented environments. In *Ninth International Conference on Computer Aided Design and Computer Graphics*, 2005.