Smooth and efficient zooming and panning

Jarke J. van Wijk Wim A.A. Nuij
Technische Universiteit Eindhoven [*]

Abstract

Large 2D information spaces, such as maps, images, or abstract visualizations, require views at various level of detail: Close ups to inspect details, overviews to maintain (literally) an overview. Users often switch between these views. We discuss how smooth animations from one view to another can be defined. To this end, a metric on the effect of simultaneous zooming and panning is defined, based on an estimate of the perceived velocity. Optimal is defined as *smooth and efficient*. Given the metric, these terms can be translated into a computational model, which is used to calculate an analytic solution for optimal animations. The model has two free parameters: animation speed and zoom/pan trade-off. A user experiment to find good values for these is described.

CR Categories: I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques; D.2.2 [Software Engineering]: Tools and Techniques—User interfaces

Keywords: Navigation, zooming, panning, scrolling, scale space

1 INTRODUCTION

We consider a simple problem. Suppose, we are developing an interactive cartographic application. The user is presented a map of, say, the US, and can zoom in on regions, states, and cities by picking items from a list or clicking on areas on the screen. We want to offer a smooth animation from one close-up on the map to another. How to define this animation?

One encounters this problem frequently. Cartography is the prime example, but in Information Visualization there is an even stronger need for such smooth animations. Abstract data is typically mapped to 2D graphic representations, such as scatterplots, graph diagrams, or treemaps. Large data sets lead to large images with much detail. Interaction with these representations is vital to achieve insight, as summarized in Ben Shneiderman's Visual Information Seeking mantra: *Overview, zoom & filter, details-on-demand* [Shneiderman 1996]. Another key concept in Information Visualization is *focus+context*. Both stress that data must be visualized at several levels of scale: The user must be enabled to zoom in and focus, while on the other hand he should maintain an overview and understand the context of the data focussed on. One solution is to offer multiple representations at different scales simultaneously, another class of solutions concerns distortion of space, such as fisheye views [Furnas 1986]. Here we consider the use of the time dimension for this purpose. In other words, if the user shifts his attention, from overview to detail or from one detail to another, a smooth transition aids in understanding the relation between the two views.

At first sight, interpolation (linear in space, logarithmic in scale) might seem to be sufficient to make the transition from one view to another. However, this solution falls short when the transition has to be made from one close-up to another. For instance, suppose we focus on New York and shift to Los Angeles. Such a simple solution leads to a long animation, where a small strip of the US is shown in detail. A somewhat better solution is to zoom out first, pan across the continent, followed by a zoom in on the city of destination. But how much to zoom out? How much time should the animation take? How to combine zooming and panning? What is the optimal path? How can we define optimal here? The problem is less simple than it seems at first sight.

In this paper we present a solution to this problem, or, in other words, we present a computational model for fast navigation in scale space. After a review of related work in section 2, we analyze the problem in section 3. Central is the definition of a metric on the effect of zooming and panning, derived from an estimate of average velocity. Based on this metric we first present an optimal solution for a simple zoom-out, pan, zoom-in scenario (section 4). Next we consider arbitrary transitions (section 5) and present how an optimal path of a virtual camera can be determined analytically given two projections. In section 6 we present a first user experiment in order to find satisfying values for the two free parameters in the model (animation speed and zoom/pan trade off). Finally, conclusions are drawn and possible extensions are discussed in section 7.

2 BACKGROUND

The importance of viewing at different levels of scale, or for short *multiscale viewing*, is addressed in many articles on visualization, and can be found in many interactive applications, not only visualization tools, but also image viewers, word processors, games, etc.. Bederson has shown with his work on Zoomable User Interfaces [Bederson et al. 1994; Bederson et al. 2000] how flexible viewing can be used as a foundation for intuitive user interfaces.

Surprisingly however, we could only find one reference where the problem we address here is discussed explicitly. George Furnas and Ben Bederson [1995] present Space-Scale Diagrams: a visual depiction of multiscale viewing. The horizontal axis denotes space, the vertical axis denotes scale. Furnas and Bederson show how this diagram can be employed to attack a variety of problems associated with multi scale viewing, including optimal pan-zoom trajectories. Optimal is translated here as the shortest path. To determine the length of a path, they present measures for pure panning and zooming, based on an information metric. The distance between two views is expressed as the number of bits required to encode the difference between the two frames. For panning this is equal to βd, where β is the bit density of the image and d is the displacement in screen units, for zooming this is equal to $\beta w \log r$, where w is the width of the screen, and r is the zoom factor. Guidelines for several piecewise pure pans or zooms trajectories are given, but for the truly optimal shape, the authors remark that such trajectories *"will*

[*] {vanwijk, wsinwaan}@win.tue.nl, Department of Mathematics and Computer Science, Technische Universiteit Eindhoven, The Netherlands

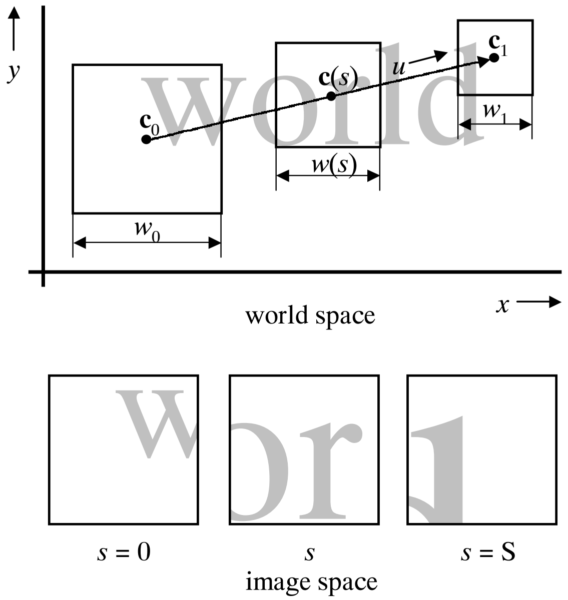

Figure 1: World space and image space

have a complicated curved shape, and finding it is a complicated calculus-of-variations problem".

A related topic has been addressed by Igarashi *et al.* [2000]. They consider how to improve the browsing through large documents. They recommend that during scrolling the view should zoom out automatically, such that the perceptual scrolling speed in screen space remains constant.

Our approach has a similar structure as that of Furnas and Bederson, but differs in the way various aspects are filled in. We discuss the problem in u, w space, where u denotes panning and w denotes zooming, both measured in world coordinates. We use u, w diagrams, which are simpler to understand and work with than space-scale diagrams. We define optimal as *smooth* and *efficient*, define a metric to measure the effect of combined zooming and panning, and derive, based on differential geometry, differential equations that describe optimal paths. We solve these explicitly, first for a zoom-pan-zoom strategy, and next to find the optimal path. Finally, we present a small-scale user experiment to find good values for the two free parameters we have introduced.

3 MODEL

In this section we define the projection in more detail, followed by our definition of optimality, and the definition of a metric to assess the effect of zooming and panning. This metric is next used to define optimality quantitatively.

3.1 Projection

We consider the projection of a square subset A of an object M to image space I. M is defined over an area $W \subset \mathbb{R}^2$, $A \subset W$, where W denotes world space. We define the area of interest A by a center point $\mathbf{c} = (c_x, c_y) \in W$ and a width w, i.e.

$$A = [c_x - w/2, c_x + w/2] \times [c_y - w/2, c_y + w/2].$$

For the image space I we use normalized device coordinates: a unit square, centered around the origin, i.e.

$$I = [-1/2, 1/2] \times [-1/2, 1/2].$$

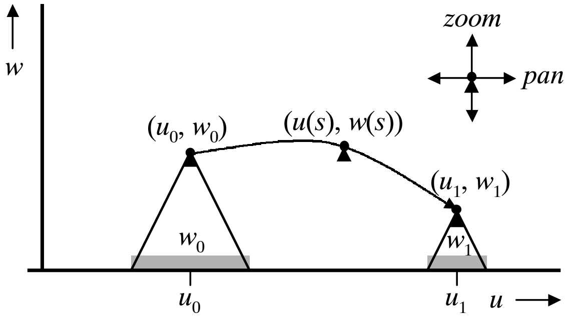

Figure 2: u, w space diagram

The projection of a point $\mathbf{x} = (x, y) \in A$ to a point $\mathbf{x}' = (x', y') \in I$ is then simply

$$(x', y') = \left(\frac{x - c_x}{w}, \frac{y - c_y}{w} \right), \tag{1}$$

the inverse projection is given by

$$(x, y) = (c_x + wx', c_y + wy'). \tag{2}$$

Suppose now that two areas of interest A_0 and A_1 are given, defined by $\mathbf{c}_i, w_i$, with $i = 0, 1$, and that an animation from the first to the second has to be defined (see figure 1). To this end we have to find functions $\mathbf{c}(s)$ and $w(s)$, $s \in [0, S]$, such that at least

$$\begin{aligned} \mathbf{c}(0) &= \mathbf{c}_0, \quad & w(0) &= w_0, \\ \mathbf{c}(S) &= \mathbf{c}_1, \quad & w(S) &= w_1. \end{aligned}$$

The parameter s is along a path from the first to the second projection, and S denotes the final value. The functions $\mathbf{c}(s)$ and $w(s)$ denote the path of the camera and the width shown along the path. An animation can now be defined by setting

$$s = Vt, \ t \in [0, S/V], \tag{3}$$

where V denotes the constant animation speed, and t wall clock time, for instance in seconds. In the remainder we will use s as main parameter, decoupled from the basic animation speed V. For convenience, if a unit speed is assumed, s and t are interchangeable.

We simplify the problem by assuming that there is no perceptual difference between horizontal, diagonal, and vertical panning. In this case an optimal path $\mathbf{c}(s)$ is always a straight line, and hence we can define

$$\begin{aligned} &\mathbf{c}(s) = \mathbf{c}_0 + \frac{\mathbf{c}_1 - \mathbf{c}_0}{\|\mathbf{c}_1 - \mathbf{c}_0\|} u(s), \ u \in [u_0, u_1], \\ &\text{with } u_0 = 0 \text{ and } u_1 = \|\mathbf{c}_1 - \mathbf{c}_0\|. \end{aligned} \tag{4}$$

The parameter $u(s)$ denotes panning along a straight line. We have to find functions $u(s)$ and $w(s)$, $s \in [0, S]$, such that at least

$$\begin{aligned} u(0) &= u_0, \quad & w(0) &= w_0, \\ u(S) &= u_1, \quad & w(S) &= w_1. \end{aligned} \tag{5}$$

We can depict the problem in (u, w) space, as shown in figure 2. We found these diagrams, which we called u, w-diagrams, more convenient to reason about the problem than space scale diagrams. Each projection maps to a point in the diagram. Zooming and panning naturally map to moving a point vertically or horizontally. The axes have the same dimension, both units in world space. Furthermore, the diagram enables a direct physical interpretation. The horizontal axis can be considered as a cross-section through the object

M to be displayed, the point can be interpreted as a camera, floating at a height w above M with a field of view of $2 \arctan 1/2 \approx 53^\circ$. The path $(u(s), w(s))$ is hence simply the path of a camera, flying above a map. This can be emulated physically: Position yourself in front of an image, look perpendicular to this image, and move your head according to a camera path (or, move the image). In this way we can try out various scenarios for zooming and panning in an easy way, and obtain a rough feeling for optimal paths.

3.2 Requirements

The next question is what an optimal path is. We think this can be summarized in two words: The optimal path should be *smooth* and *efficient*. Smoothness is a constraint. The path should be at least continuous in the first order, in the sense that no sudden steps are made or abrupt changes in direction occur. These are requirements on the shape of the path. Furthermore the parametrization of the path must be chosen carefully. We formulate this as follows. When the camera moves along the path, the viewer should get the impression of a uniform and constant motion of the projected image on the screen. We limit ourselves to the perceptual level here, and discard cognitive aspects, such as memory, meaning of the image shown etc. Such aspects are much harder to incorporate in a model, and we assume that a perceptually smooth motion will also aid in cognition.

Furthermore, in the remainder we assume that each projected part of M has the same characteristics, i.e. each image shown is equally interesting, has the same visual density, etc. for the range of $\mathbf{c}$ and w of interest, and that there are no discontinuities in the displayed image for varying w. In a strict sense, this requirement can only be met by artificial imagery with fractal characteristics. In real world applications, such as cartography, urban areas are more interesting than uniformly colored oceans, in strong close-up views often less detail is available. On the other hand, most applications for which interactive zooming is interesting will have details at many scales, and it is the task of the designer to make sure that at each scale an appropriate level of detail is shown. In cartography this is well-known as generalization.

The aspect to be optimized is efficiency. We operationalize this by aiming for the shortest path in u, w space possible: Detours are not appreciated, we want to get from A to B as fast as possible.

3.3 Metric

We aim for a path that is smooth and efficient. Both require that we are able to measure the effect of changing $\mathbf{c}$ and w, as perceived by the viewer. Following and generalizing the approach of Igarashi and Hinckley [2000], we use the velocity of the moving image as a basis for measurements, i.e., we aim at a metric for the perceived average optic flow in the image window. To this end, we first consider the velocity $\dot{\mathbf{x}}'$ of a projected point $\mathbf{x}'$ in image space. We use a centered dot as notation for differentiation with respect to s, e.g. $\dot{a} = da/ds$. Differentiation of (1) gives

$$\dot{\mathbf{x}}' = \left(\frac{-x\dot{w} - \dot{c}_x w + c_x \dot{w}}{w^2}, \frac{-y\dot{w} - \dot{c}_y w + c_y \dot{w}}{w^2} \right),$$

or, using (2)

$$\dot{\mathbf{x}}' = \left(\frac{-x'\dot{w} - \dot{c}_x}{w}, \frac{-y'\dot{w} - \dot{c}_y}{w} \right).$$

We are not only interested in a single point, we have to measure the velocity over the whole screen space I. For this we use the root mean squared average velocity V_{RMS}

$$\begin{aligned} V^2_{\mathrm{RMS}} &= \frac{V^2 \int_I \dot{\mathbf{x}}' \cdot \dot{\mathbf{x}}' dI}{\int_I dI} \\ &= V^2 \int_{-1/2}^{1/2} \int_{-1/2}^{1/2} (\dot{x}'^2 + \dot{y}'^2) dx' dy' \\ &= V^2 \left(\frac{1}{w^2} \dot{c_x}^2 + \frac{1}{w^2} \dot{c_y}^2 + \frac{1}{6w^2} \dot{w}^2 \right) \\ &= V^2 \left(\frac{1}{w^2} \dot{u}^2 + \frac{1}{6w^2} \dot{w}^2 \right). \end{aligned}$$

V_{RMS} is proportional to the animation velocity V, and the zoom velocity $\dot{w}$ and pan velocity $\dot{u}$, both relative to the width w in world space. Also, V_{RMS} shows that zooming has less impact (the factor $1/6$) than panning. At this point it is too early to use V_{RMS} directly as a measure. We cannot be sure that the perceptual effect of zooming versus panning is indeed measured by the average velocity. Hence, we define a metric on (u, w) space that is more general:

$$ds^2 = \frac{\rho^2}{w^2} du^2 + \frac{1}{\rho^2 w^2} dw^2. \tag{6}$$

This metric gives the distance ds travelled, when u and w are changed with du and dw. The parameter ρ represents a trade-off between zooming and panning. A high value indicates that zooming has little impact, a low value indicates that panning has less impact. For $\rho = 6^{1/4} \approx 1.565$, the metric is equivalent to using V_{RMS} as a measure. The best value for ρ depends on the subjective perception of the viewer, and has to be found experimentally. Results from such experiments and suggestions for good values of ρ are given in section 6.

Figure 3 shows a visualization of the metric defined in (6) for various values of ρ in (u, w) space. Each small ellipse denotes a set of points equidistant to its center according to the metric in (u, w) space. The shape of the ellipses is determined by ρ, their size is proportional to w.

Given the metric, an optimal path can now be defined more precisely. Two conditions must be satisfied. First, the animation should be smooth. In other words, when s varies constantly, the perceived rate of change has to be constant according to the metric. This implies that the path $(u(s), w(s))$ has to be arc length parametrized, and should satisfy the following differential equation, derived directly from the metric:

$$\rho^2 \dot{u}^2 + \dot{w}^2/\rho^2 = w^2. \tag{7}$$

Secondly, the animation should be efficient. If the path is arc length parametrized, then s represents the distance travelled. Efficiency then implies that the total distance S should be minimal.

In terms of figure 3, the task of finding an optimal path loosely comes down to finding a path between two points, such that at each step along the path the same number of ellipses is crossed, and that in total as few ellipses as possible are crossed. For each value of ρ such an optimal path is shown, where the dots indicate equidistant points. The shapes of these paths depend on ρ: little zooming for low ρ, much zooming for high ρ.

4 ZOOM OUT, PAN, ZOOM IN

We illustrate the preceding ideas first for a simple scenario, in the next section we consider optimal paths. Figure 4 shows a simple path:

- for $s = 0$ to s_A: zoom out from (u_0, w_0) to (u_0, w_m);

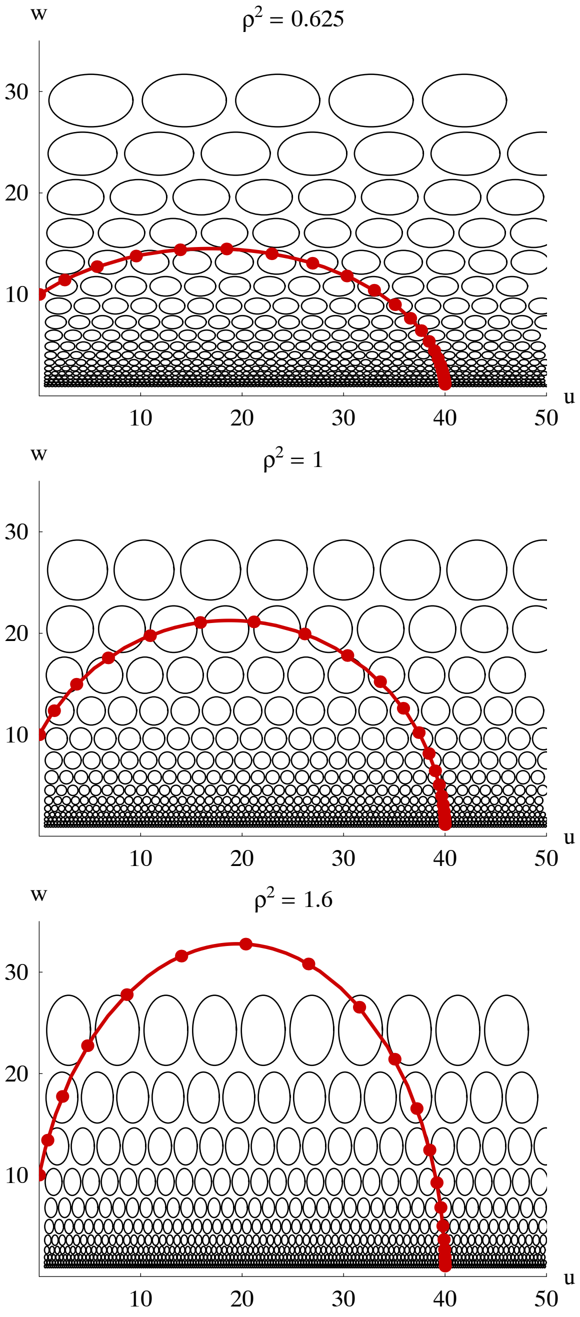

Figure 3: Metric in (u, w) space

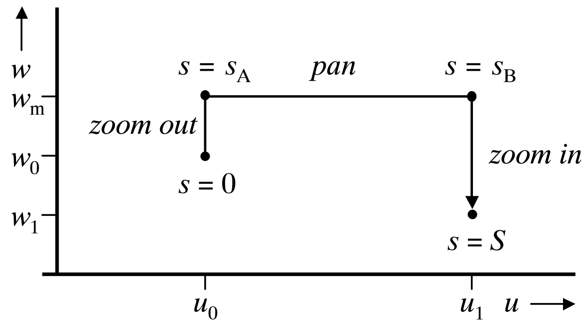

Figure 4: Zoom out, pan, zoom in

- for $s = s_A$ to s_B: pan from (u_0, w_m) to (u_1, w_m);
- for $s = s_B$ to S: zoom in from (u_1, w_m) to (u_1, w_1).

The problem now is to define a path $(u(s), w(s))$ such that the path is arc length parametrized and that the total path length S is minimal. For the first interval, where $\dot{u} = 0$, the first condition (7) reduces to $\dot{w} = \rho w$. A solution of this equation that satisfies $w(0) = w_0$ is

$$w(s) = w_0 \exp(\rho s), \; s \in [0, s_A].$$

This is consistent with the recommendation given in [Mackinlay et al. 1990], where the use of a logarithmically slower movement is advocated when the target (in 3D) is approached. Similarly we find for the last interval

$$w(s) = w_m \exp(\rho(s_B - s)), \; s \in [s_B, S].$$

For the panning interval in between, where $\dot{w} = 0$, the arc length parametrization condition reduces to $\dot{u} = w_m/\rho$. The solution of this equation that satisfies $u(s_A) = u_0$ is

$$u(s) = w_m(s - s_A)/\rho + u_0, \; s \in [s_A, s_B].$$

The quantities s_A, s_B, and S depend on w_m. The value of w_m should be chosen such that S is minimal. We determine first the total length by considering the length of the three intervals. First, the length of the first interval follows from $w(s_A) = w_m$, i.e.

$$s_A = \ln(w_m/w_0)/\rho, \; w_m \geq w_0.$$

Similarly, we find for the length of the last interval

$$S - s_B = \ln(w_m/w_1)/\rho, \; w_m \geq w_1.$$

The constraints $w_m \geq w_0$ and $w_m \geq w_1$ are required here to make sure that no negative pathlengths are generated. Using $u(s_B) = u_1$ we find for the length of the panning phase:

$$s_B - s_A = \rho(u_1 - u_0)/w_m.$$

Hence, the total length S, expressed in w_m, is

$$S = \frac{\rho(u_1 - u_0)}{w_m} + \frac{\ln(w_m/w_0)}{\rho} + \frac{\ln(w_m/w_1)}{\rho}.$$

S is minimal if $dS/dw_m = 0$. This gives

$$\frac{-\rho(u_1 - u_0)}{w_m^2} + \frac{2}{\rho w_m} = 0.$$

Together with the constraints that $w_m \geq w_0$ and $w_m \geq w_1$, we get the optimal value for w_m

$$w_m = \max(w_0, w_1, \rho^2(u_1 - u_0)/2),$$

which completes the derivation of the optimal path for this simple scenario. In summary, we have found that the optimal path $(u(s), w(s))$ for $s \in [0, S]$ for the zoom-pan-zoom scenario is given by:

$$\boxed{\begin{array}{rcl}
u(s) & = & \begin{cases} u_0 & \text{if } 0 \leq s < s_A \\ w_m(s - s_A)/\rho + u_0 & \text{if } s_A \leq s < s_B \\ u_1 & \text{if } s_B \leq s \leq S \end{cases} \\
w(s) & = & \begin{cases} w_0 \exp(\rho s) & \text{if } 0 \leq s < s_A \\ w_m & \text{if } s_A \leq s < s_B \\ w_m \exp(\rho(s_B - s)) & \text{if } s_B \leq s \leq S \end{cases} \\
s_A & = & \ln(w_m/w_0)/\rho \\
s_B & = & s_A + \rho(u_1 - u_0)/w_m \\
S & = & s_B + \ln(w_m/w_1)/\rho \\
w_m & = & \max(w_0, w_1, \rho^2(u_1 - u_0)/2).
\end{array}}$$

Ignoring the constraints, the optimal value of w_m depends only on ρ and on the distance between $\mathbf{c}_0$ and $\mathbf{c}_1$. For $\rho = 6^{1/4}$, $w_m \approx 1.2\|\mathbf{c}_1 - \mathbf{c}_0\|$, i.e. one has to zoom out to such a level that both the start point $\mathbf{c}_0$ and the end point $\mathbf{c}_1$ are visible at some moment during the flight.

5 THE OPTIMAL PATH

The preceding section dealt with a path with a rectangular shape. The smoothness criterion is violated here: At the corners the motion is discontinuous. Also, the total length is not optimal: By cutting corners a shorter path can be achieved. An optimal path between (u_0, w_0) and (u_1, w_1) has to satisfy the boundary conditions given in (5), and the arc length parametrization condition (7). Furthermore, it has to be the shortest path between the two points, i.e. the *geodesic*.

In standard Euclidean space a geodesic is a straight line, in curved space this is usually a curve. A classic example is spherical space, used for mapping spheres. On a longitude, latitude map a line does not give the shortest path on the sphere, whereas a great circle does. Also our (u, w) space is curved, because of the metric we have defined. In Appendix 1 we illustrate how this curved space can be depicted geometrically.

Curved space is studied with analytical means in differential geometry, of which Gauss and Beltrami can be considered the founders. From this vast area we only need how from the metric an equation for the geodesics can be found [Struik 1950; Auslander 1967; Lipschutz 1969]. Differential geometry tells us that for a space with a metric of the form

$$ds^2 = E\,du^2 + G\,dw^2,$$

a geodesic $(u(s), w(s))$ has to satisfy the following equations

$$\begin{aligned}
\ddot{u} + \frac{E_u}{2E}\,\dot{u}^2 + \frac{E_w}{E}\,\dot{u}\dot{w} - \frac{G_u}{2E}\,\dot{w}^2 = 0, \text{ and} \\
\ddot{w} - \frac{E_w}{2G}\,\dot{u}^2 + \frac{G_u}{G}\,\dot{u}\dot{w} + \frac{G_w}{2G}\,\dot{w}^2 = 0,
\end{aligned}$$

where double dots denote double differentiation with respect to s (e.g. $\ddot{u} = d^2u/ds^2$), and subscripts denote partial differentation (e.g. $E_u = \partial E/\partial u$). For our metric $E = \rho^2/w^2$ and $G = 1/\rho^2 w^2$, substitution gives

$$\begin{aligned}
\ddot{u} - 2\dot{u}\dot{w}/w = 0, \text{ and} \\
\ddot{w} + \rho^4\dot{u}^2/w - \dot{w}^2/w = 0.
\end{aligned} \tag{8}$$

Hence, the optimal path is the solution for $(u(s), w(s))$ that satisfies equations (5), (7), and (8). In Appendix 2 we show how the analytical solution can be derived, here we give the final result for $(u(s), w(s))$, $s \in [0, S]$, $u_0 \neq u_1$:

$$\boxed{\begin{array}{rcl}
u(s) & = & \dfrac{w_0}{\rho^2}\cosh r_0 \tanh(\rho s + r_0) - \dfrac{w_0}{\rho^2}\sinh r_0 + u_0, \\
w(s) & = & w_0 \cosh r_0 / \cosh(\rho s + r_0), \\
S & = & (r_1 - r_0)/\rho, \\
r_i & = & \ln(-b_i + \sqrt{b_i^2 + 1}),\ i = 0, 1, \text{ and} \\
b_i & = & \dfrac{w_1^2 - w_0^2 + (-1)^i \rho^4 (u_1 - u_0)^2}{2 w_i \rho^2 (u_1 - u_0)},\ i = 0, 1,
\end{array}} \tag{9}$$

where the hyperbolic cosine, sine, and tangens are defined as $\cosh x = (e^x + e^{-x})/2$, $\sinh x = (e^x - e^{-x})/2$, and $\tanh x = \sinh x / \cosh x$. For $u_0 = u_1$ the optimal path is given by

$$\boxed{\begin{array}{rcl}
u(s) & = & u_0 \\
w(s) & = & w_0 \exp(k\rho s) \\
S & = & |\ln(w_1/w_0)|/\rho \\
k & = & \begin{cases} -1 & \text{if } w_1 < w_0 \\ 1 & \text{otherwise} \end{cases}
\end{array}}$$

Figure 5 shows sets of geodesic paths, starting from $u = 0$ and $w = 10$ in different directions, for various values of ρ. Furthermore, in each plot a set of contours is shown as thin lines. Each contour represents a set of points at an equal distance from the start point. Both the paths and the contours are parts of ellipses, where ρ again determines their shapes. A path lies on the ellipse through (u_0, w_0) and (u_1, w_1) with center $(u_0 - w_0 \sinh r_0/\rho^2, 0)$, the lengths of the axes in u and w direction are $w_0 \cosh r_0/\rho^2$ and $w_0 \cosh r_0$ respectively. For $\rho = 1$ paths are circles. The center lies on the u-axis. For $s \to \pm\infty$ the path approaches the horizontal axis perpendicularly, i.e. for small w panning is not effective.

Given these images, the structure of equation (9) can now be understood more easily. The elliptic path is defined by $(u, w) = (\tanh(s)/\rho^2, 1/\cosh(s))$, the other constants involved make sure that the ellipse passes through (u_0, w_0) and (u_1, w_1), and that the parametrization fits the measure.

Figure 6 shows another visualization of the paths. The horizontal axis denotes s, the vertical axis u. The thick lines show $u(s)$ for three values of ρ. This shows that the virtual camera moves smoothly and monotonically in the direction of u_1. Furthermore the instantaneous width is shown: the interval between the graphs of $u(s) - w(s)/2$ and $u(s) + w(s)/2$, shown as thin lines. Again, the effect of different values for ρ is clearly visible.

The scale of s has not been discussed so far. For this case, the total path length S varies from about 4.2 to 4.5. What is this dimension? For a pure panning motion we find that value s corresponds to a motion of $\rho\Delta u/w$, i.e. s relates here to ρ times the number of image widths panned, which is a fairly natural and understandable measure. For example, for $\rho = 1$ panning from $(u, w) = (0, 10)$ to $(40, 10)$ gives a distance $s = 4$. For a zoom out with a factor

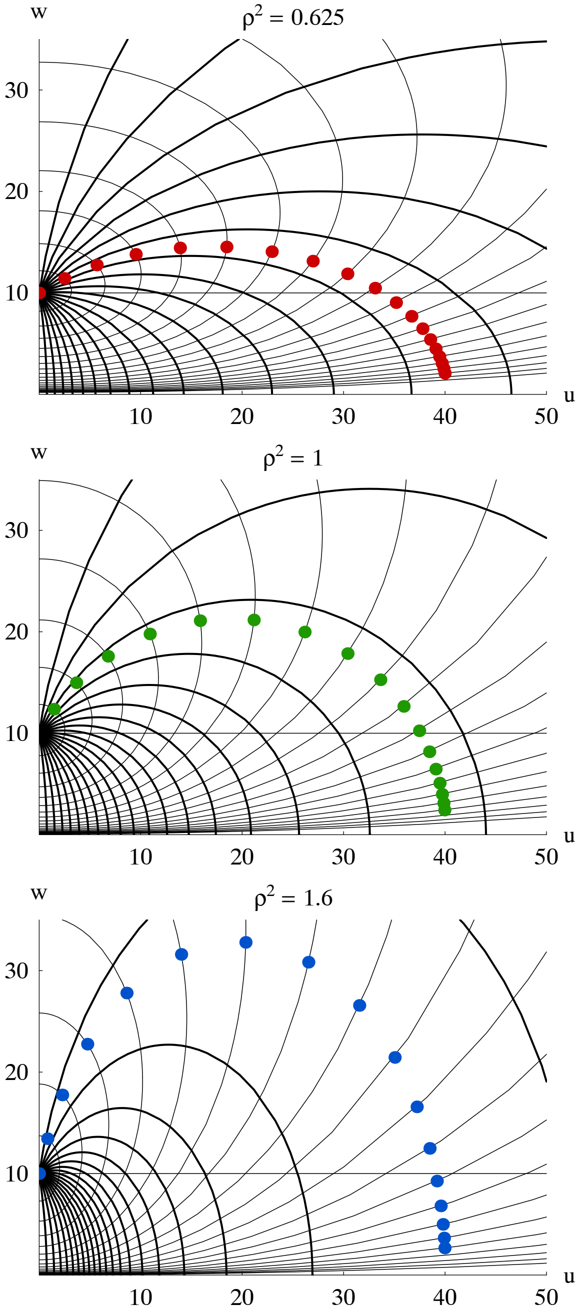

Figure 5: Paths and iso-distance curves in (u, w) space

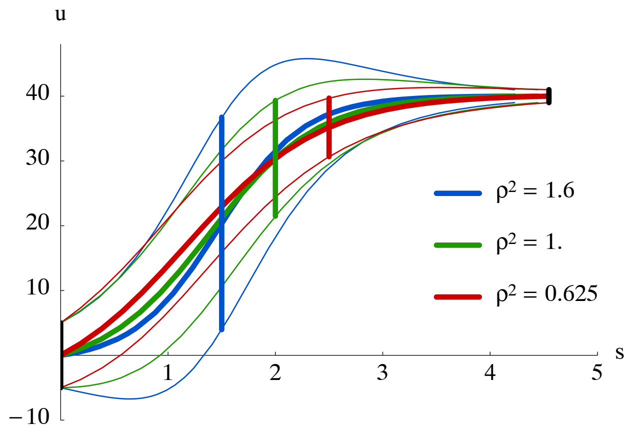

Figure 6: u and w as a function of s

$r = w_1/w_0$ we find that the corresponding s equals $\ln r/\rho$. Hence, zooming in from (40, 10) to (40, 1) gives a distance of $s \approx 2.3$.

The paths derived here and in the previous section can easily be translated into an implementation of a smooth animation. The simplest and also most flexible approach is to calculate for each new frame everything anew. We take advantage here of the property that geodesics are unique: if C is a point on the geodesic from A to B, then the geodesic from C to B is a subset of the original one. We typically use a procedure which takes as input the current view ($\mathbf{c}_0$ and w_0), the target view ($\mathbf{c}_1$ and w_1), the animation parameters (V and ρ), and the time step Δt between frames, and which returns the view ($\mathbf{c}$ and w) for the next frame. For the time step we use an average value of the last five to ten frames.

Another implementation is to recalculate the path only when one of the parameters changes. This requires two separate procedures (one for making a step and one for changing the path), where care has to be taken that the latter one is always called when a parameter is changed. The suggested state-less implementation takes care of this automatically, simplifying the change of parameters during the flight, at the expense of some additional computing cost. But typically, this cost is neglectable when compared to the rendering of the frame.

6 USER EXPERIMENTS

We have done a first user experiment to obtain insight in preferred values for V and ρ. We have implemented a small application where the user can load an image, define areas of interest, switch between these areas, upon which a smooth animation is shown with user-defined settings for V and ρ. Also, the application has a test mode for more controlled experiments. We used a high resolution height map of Mars as image. This image was visually interesting, contains information on various scales, and was unfamiliar to our users. We explained the background of the experiments, and asked the users to set V and ρ for two conditions. First, an alternating zoom-pan animation between two locations was shown, secondly, a tour around ten different locations was shown. We asked the users to set the parameters such that the perceived animation was smooth, fluent, and suitable for a daily use application. The task was done at a notebook computer, and took about 5 minutes to complete. Users could set the values via sliders, no feedback on numerical values was given to prevent bias. The 26 users were colleagues and students from our department, and all had (much) experience with using computers.

Many users commented on the importance of cognitive aspects.

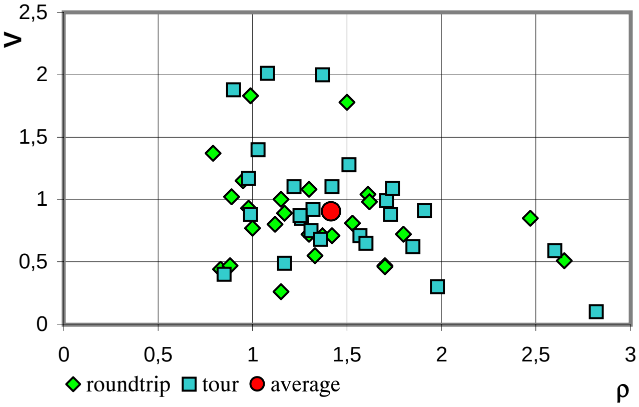

Figure 7: Scatterplot of V and ρ values found

They pointed out that familiarity of the image shown, the amount of detail in the image, and the exact task would influence their preferences. Also, we found that some users set ρ to get a zoom-out level which they liked, where the aspect of smooth motion was of lesser importance. On the positive side, the users found the final paths (given their preferences) smooth, pleasant, and natural.

Figure 7 shows the parameter settings that were selected by our users. It shows that preference for ρ and V were uncorrelated, hence these seem to be independent dimensions, and that the conditions (two vs. ten locations) did not strongly influence the result. The average value for V is 0.90, the standard deviation 0.43. For ρ the average value is 1.42 and the standard deviation is 0.47. The value of 1.42 is (statistically significant) smaller than 1.565, the value which we derived from using the root mean squared average velocity. The average value found suggests that $\rho = \sqrt{2}$ is possibly an optimal value, but we have not found yet a model to explain this. Furthermore, the variation in the results was large, much larger than we expected. Nevertheless, we think that these average values will yield reasonable results for a variety of users and use cases. For an optimal result, the setting has to be customizable to the preference of the user.

7 CONCLUSIONS

We have shown how to generate a smooth animation from one view on a 2D model to another. The problem turned out to be more complex than it seems at first sight, we needed differential geometry and had to solve a system of three differential equations to find an optimal path. Was this worth it? We think it was. The end result is a set of closed equations, which can easily be used to implement smooth and efficient zooming and panning. The visual result, as we observed from user tests is pleasing and smooth. The original impetus for this work was the visualization of call graphs via a hierarchical matrix visualization [van Ham 2003]. The use of a smooth zoom-pan strategy to maintain a good overview was vital here, because the information shown is abstract and varies per level. The method described here has been integrated, and gives very good results, visually. Finally, there are many other applications that can take advantage of this method, i.e. any application where a user observes some 2D object and is enabled to zoom and pan.

Besides from practical use, the approach described here is also interesting from a more theoretical point of view. We used measurement of average perceived velocity as our starting point, and have presented a model to relate zooming and panning. In the future we will explore if the approach described here also can deal with more general image transforms, especially anisotropic scaling and rotation. Another possible variation is to use a varying velocity, such as acceleration at the beginning and deceleration at the end of the animation. Furthermore, we focussed here on animations where the start and end are given. The metric introduced here can possibly also be used to optimize free navigation, where the user is free to control the path himself.

We aimed at perceptually pleasing results, however, cognitive aspects also play an important role for this kind of animations. More research can be done to study the role of perception versus cognition. Specifically, it is interesting to study what the average value and variation of ρ is when perceptual effects are isolated in some way. A more precise analysis of the various cognitive aspects as well as guidelines for good values for ρ given a variety of tasks, image contents, etc. would be useful. An extended model, in which the user can additionally specify that he for instance wants to zoom out more than the optimal paths defined here could also be useful to tackle the cognitive aspects. Meanwhile however, the model presented here is already useful for a variety of cases, especially if the user can tune the two parameters to his preferences.

References

Auslander, L. 1967. *Differential geometry*. Harper&Row. p. 165.

Bederson, B., Stead, L., and Hollan, J. 1994. Pad++: Advances in multiscale interfaces. In *Proceedings of CHI'94 Human Factors in Computing Systems*, 315–316.

Bederson, B., Meyer, J., and Good, L. 2000. Jazz: An extensible zoomable user interface graphics toolkit in java. In *Proceedings UIST 2000, ACM Symposium on User Interface Software and Technology*, 171–180.

Furnas, G., and Bederson, B. 1995. Space-scale diagrams: Understanding multiscale interfaces. In *Proceedings of CHI'95 Human Factors in Computing Systems*, 234–241.

Furnas, G. 1986. Generalised fisheye views. In *Proceedings of CHI'86*, 16–23.

Igarashi, T., and Hinckley, K. 2000. Speed-dependent automatic zooming for browsing large documents. *CHI Letters 2*, 2, 139–148. Proceedings UIST 2000, ACM Symposium on User Interface Software and Technology.

Lipschutz, M. 1969. *Differential Geometry*. McGraw-Hill. p. 234.

Mackinlay, J., Card, S., and Robertson, G. 1990. Rapid controlled movement through a virtual 3d workspace. *Computer Graphics 24*, 4, 171–176. Proceedings of SIGGRAPH'90.

Shneiderman, B. 1996. The eyes have it: A task by data type taxonomy of information visualizations. In *Proceedings of IEEE Symposium on Visual Languages '96*, 336–343.

Struik, D. 1950. *Lectures in classical differential geometry*. Addison-Wesley. p. 142.

van Ham, F. 2003. Using multi-level call matrices in large software projects. In *Proceedings IEEE InfoVis 2003*.

APPENDIX 1 – CURVED SPACE

To illustrate this concept of curved space, we consider what geometric surface corresponds to our metric, like a sphere corresponds with the longitude latitude map. Loosely, we distort the 2D images shown in figure 3 to a 3D surface, such that all ellipses are distorted to circles with the same size. To this end, one has to shrink horizontal lines (w constant) for high values of w, and to stretch them for low values. This will give a rotationally symmetric 3D surface. Horizontal lines are distorted into circles, or, in other words, the u parameter is mapped to rotation.

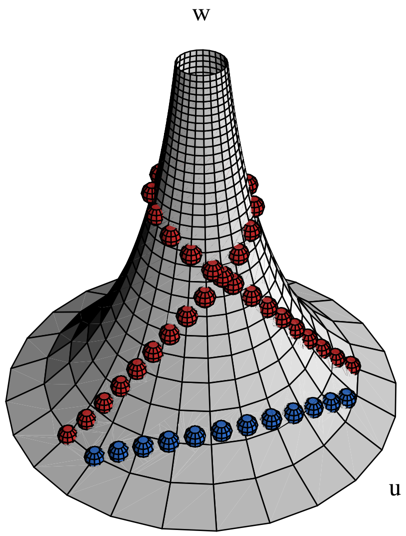

Figure 8: u, w space depicted as a geometric surface

More formally, let us consider a parametric surface $\mathbf{x}(u, w) = (x(u, w), y(u, w), z(u, v))$. The length of a small step $d\mathbf{x} = \mathbf{x}_u du + \mathbf{x}_v dv$ has to fit the measure, i.e.

$$d\mathbf{x} \cdot d\mathbf{x} = \frac{\rho^2}{w^2}\, du^2 + \frac{1}{\rho^2 w^2}\, dw^2.$$

Figure 8 shows a surface that satisfies this constraint:

$$\begin{aligned} x(u, w) &= (\rho/w) \cos u, \\ y(u, w) &= (\rho/w) \sin u, \text{ and} \\ z(u, w) &= \frac{1}{\rho} \ln(w + \sqrt{w^2 - \rho^4}) - \frac{\sqrt{w^2 - \rho^4}}{w\rho}. \end{aligned}$$

The horizontal circles are lines with constant w, the lines towards the top are lines with constant u. In other words, panning is mapped to rotating around the central axis. Panning over large distances corresponds to rotating multiple times around the central axis, each point on the surface is multi-valued. Zooming out corresponds to climbing the object, zooming in to descending. Note that the square grid cells shrink when w increases. Two geodesics are shown. For large panning distances (moving multiple times around the central axis) the geodesic is located high on the surface, i.e. zooming out is stronger.

APPENDIX 2 – DERIVATION PATH

In this appendix we give the derivation of the optimal path $(u(s), w(s))$, $s \in [0, S]$. This path has to satisfy the boundary conditions

$$\begin{array}{ll} u(0) = u_0, & w(0) = w_0, \\ u(S) = u_1, & w(S) = w_1; \end{array} \tag{10}$$

it has to be arc length parametrized:

$$\rho^2 \dot{u}^2 + \dot{w}^2/\rho^2 = w^2, \tag{11}$$

and has to satisfy the geodesic equations

$$\begin{aligned} &\ddot{u} - 2\dot{u}\dot{w}/w = 0, \text{ and} \\ &\ddot{w} + \rho^4 \dot{u}^2/w - \dot{w}^2/w = 0. \end{aligned} \tag{12}$$

Equation (11) can be satisfied by introducing an extra parameter $\beta(s)$ and setting

$$\dot{u} = w \sin(2\beta)/\rho, \text{ and} \tag{13}$$

$$\dot{w} = w\rho \cos(2\beta). \tag{14}$$

If we substitute this in (12), both equations reduce to

$$\rho\, ds = d\beta / \sin\beta \cos\beta.$$

Integration gives, with r_0 an integration constant,

$$\rho s + r_0 = \ln \tan \beta,$$

If we set $s^* = \rho s + r_0$ to shorten notation, then we get $\tan\beta = e^{s^*}$. The use of $\cos 2\beta = (1 - \tan^2\beta)/(1 + \tan^2\beta)$ in equation (14) leads to

$$\begin{aligned} \dot{w} &= \rho w(1 - e^{2s^*})/(1 + e^{2s^*}) \\ &= -\rho w \sinh s^* / \cosh s^*. \end{aligned}$$

Rearrangement and integration give

$$\ln w = \ln A - \ln\cosh s^*,$$

hence

$$w(s) = A / \cosh(\rho s + r_0), \tag{15}$$

where A is an integration constant.
Using $\sin 2\beta = \tan\beta/(1 + \tan^2\beta)$ in (13) a similar derivation leads to

$$u(s) = A \tanh(\rho s + r_0)/\rho^2 + B, \tag{16}$$

where B is yet another integration constant.

We have now found $u(s)$ and $w(s)$, the next step is to set the four unknown constants A, B, r_0, and S such that the boundary conditions (10) are satisfied. From $w(0) = w_0$ we get

$$A = w_0 \cosh r_0.$$

Let $r_1 = r_0 + \rho S$. From $w(S) = w_1$ we get

$$A = w_1 \cosh r_1.$$

Combination of the last two equations gives

$$w_1 \cosh r_1 - w_0 \cosh r_0 = 0. \tag{17}$$

Similarly, the use of $u(0) = u_0$ and $u(S) = u_1$ gives

$$\begin{aligned} B &= u_0 - (w_0/\rho^2) \sinh r_0, \text{ and} \\ B &= u_1 - (w_1/\rho^2) \sinh r_1, \end{aligned}$$

and combined

$$w_1 \sinh r_1 - w_0 \sinh r_0 = (u_1 - u_0)\rho^2. \tag{18}$$

Let $U = (u_1 - u_0)\rho^2$. The unknown r_1 can be eliminated from (17) and (18) by using $\cosh^2 r_1 - \sinh^2 r_1 = 1$, resulting in

$$w_1^2 = w_0^2 - 2w_0 U \sinh r_0 - U^2,$$

so

$$\sinh r_0 = (w_0^2 - w_1^2 - U^2)/2w_0 U = -b_0,$$

hence

$$r_0 = \ln(-b_0 + \sqrt{b_0^2 + 1}).$$

Similarly, elimination of r_0 from (17) and (18) gives

$$r_1 = \ln(-b_1 + \sqrt{b_1^2 + 1})$$

which completes the derivation of $u(s)$ and $w(s)$.