A Model for Smooth Viewing and Navigation of Large 2D Information Spaces

Jarke J. van Wijk and Wim A.A. Nuij

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 change their view during a session. Smooth animations enable the user to maintain an overview during interactive viewing and to understand the context of separate views. We present a generic model to handle smooth image viewing. The core of the model is a metric on the effect of simultaneous zooming and panning, based on an estimate of the perceived velocity. Using this metric, solutions for various problems are derived, such as the optimal animation between two views, automatic zooming, and the parametrization of arbitrary camera paths. Optimal is defined here as *smooth and efficient*. Solutions are based on the shortest paths of a virtual camera, given the metric. The model has two free parameters: animation speed and zoom/pan trade off. A user experiment to find good values for these is described. Finally, it is shown how the model can be extended to deal also with rotation and nonuniform scaling.

Index Terms—Navigation, viewing, zooming, panning, scrolling, scale space.

1 Introduction

The use of two-dimensional images for the presentation of information is standard and ubiquitous. Cartography is the prime example, but it is also common practice in information visualization: Abstract data are mapped to 2D graphic representations, such as scatterplots, graph diagrams, or treemaps. Large data sets lead to large images with much detail. It is vital to achieve insight into interaction with these representations, as summarized in Shneiderman's Visual Information Seeking mantra: *Overview, zoom & filter, details-on-demand* [13]. 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 enable to zoom in and focus, while, on the other hand, he should maintain an overview and understand the context of the data focused on. One solution is to offer multiple representations at different scales simultaneously, another class of solutions concerns distortion of space, such as fish-eye views [6]. Here, we consider the use of the time dimension for this purpose. Modern computer hardware enables the user to inspect, navigate, and view large images interactively and dynamically. While viewing images, when the user shifts his attention, from overview to detail or from one detail to another, smooth transitions aid in understanding the relation between different views.

We present a computational model for the definition of smooth animations for interactive image viewing and navigation. This work is an extension of a previous publication [16], where we focused on a single problem: the definition of smooth animations from one view to another. We repeat the ideas and solutions developed there, though in a slightly more compact form, and show that these can be used for other viewing applications as well.

The problem that we use as a starting point is the following: 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. How do we define a smooth animation from one close-up on the map to another?

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? This problem turns out to be less simple than it seems at first sight.

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 present (Section 4) an optimal solution for a simple zoom-out, pan, zoom-in scenario. Next, we consider arbitrary transitions and present how an optimal path of a virtual camera can be determined analytically given two projections. A first user experiment is presented in order to find satisfying values for the two free parameters in the model (animation speed and zoom/pan trade off).

Animation from one view to another is just one aspect of interactive viewing in general, albeit a crucial one. In this

• *The authors are with the Department of Mathematics and Computer Science, Technische Universiteit Eindhoven, Den Dolech 2, 5612 AZ Eindhoven, The Netherlands.*
E-mail: {vanwijk, wsinwaan}@win.tue.nl.

Manuscript received 29 Sept. 2003; revised 10 Nov. 2003; accepted 18 Nov. 2003.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCGSI-0090-0903.

extended publication, we show how these ideas can also be used to solve other problems related to interactive viewing of images. Specifically, in Section 5, we discuss how automatic zooming can be defined and how arbitrary paths can be dealt with. Furthermore, in Section 6, it is shown how the model can be extended to deal with rotation and nonuniform scaling.

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 et al. have shown with their work on Zoomable User Interfaces [2], [3] how flexible viewing can be used as a foundation for intuitive user interfaces.

Surprisingly however, we could only find a few references where the topic we address here is discussed explicitly. Furnas and Bederson [7] 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 multiscale 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 have a complicated curved shape, and finding it is a complicated calculus-of-variations problem."*

Igarashi and Hinckley [8] have considered how to improve 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.

At some more distance, other related work can be found in the field of multiscale analysis of images where differential geometry is often used. In particular, Eberly has defined a similar metric as we have to describe the effect of changes in position and scale [4]. He uses this metric to model anisotropic diffusion.

Our approach has a similar structure as that of Furnas and Bederson, but differs in the way various aspects are filled in. We discuss viewing 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. A user study has been done to obtain good values for the two parameters of the model and, finally, we apply and extend the base model for a variety of applications.

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.,

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

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

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

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 Fig. 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 of this paper, we will use s as the 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

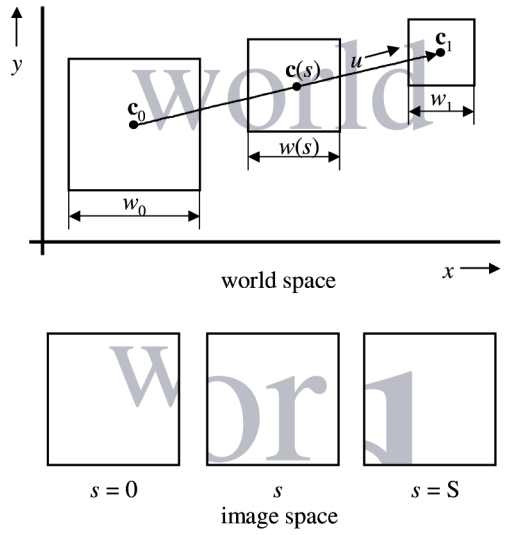

Fig. 1. World space and image space.

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

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 Fig. 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 $\mathbf{p} = (u, w)$ (or $(\mathbf{c}, w)$) can be interpreted as a camera, floating at a height w above M. The field of view of this camera is, given the definitions used here, $2 \arctan 1/2 \approx 53°$. The path $\mathbf{p}(s) = (u(s), w(s))$ 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 is an optimal path. 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 smooth and continuous 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.

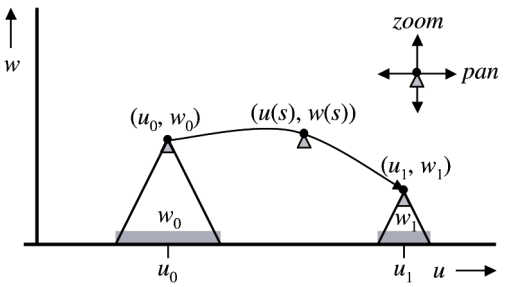

Fig. 2. u, w space diagram.

Furthermore, in the remainder of this paper, 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 [8], 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}

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

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. For zooming, this is consistent with the recommendation given in [12], where the use of a logarithmically slower movement is advocated when the target (in 3D) is approached. 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 traveled 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 4.3.

Fig. 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. Or, more technically, each ellipse visualizes a covariant metric tensor. 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:

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

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

In terms of Fig. 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.

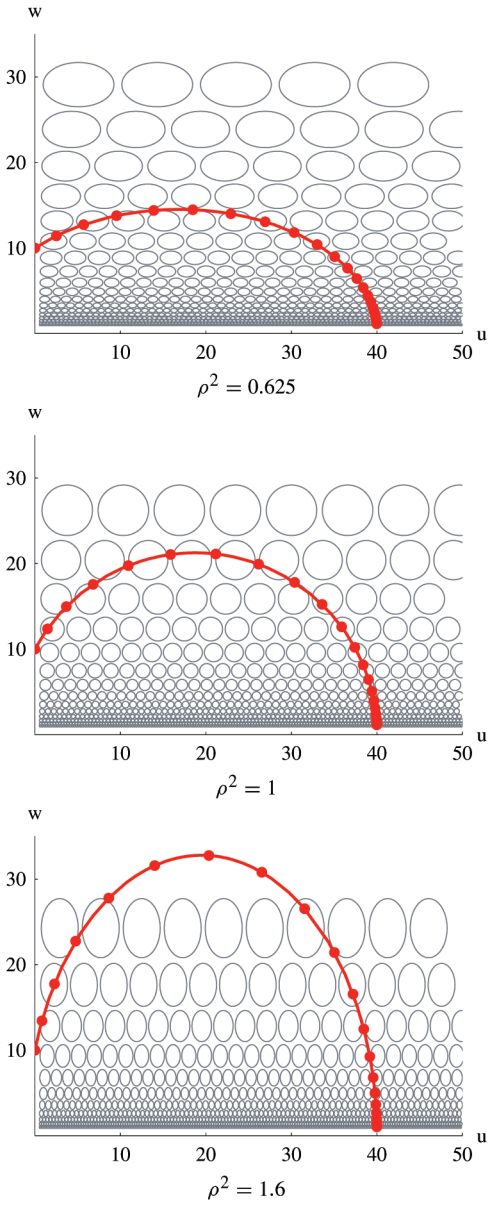

Fig. 3. Metric in (u, w) space.

4 Paths

We illustrate the preceding ideas first for a simple scenario, next we consider optimal paths, followed by a description of a user study we did to find optimal values for the parameters. In the next section, we describe a number of other applications.

4.1 Zoom Out, Pan, Zoom In

Fig. 4 shows a simple path:

- For $s = 0$ to s_A: Zoom out from (u_0, w_0) to (u_0, w_m);
- 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. In [16], we derived that the optimal path $(u(s), w(s))$ for $s \in [0, S]$ for this zoom-pan-zoom scenario is given by:

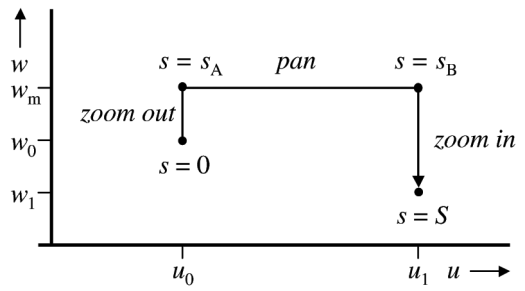

Fig. 4. Zoom out, pan, zoom in.

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

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.

4.2 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. Specifically, we are dealing here with hyperbolic space. In the Appendix, we give some more background on this.

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 to know how from the metric an equation for the geodesics can be found [1], [11], [14]. 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 (5), (7), and (8). In Appendix 2 of [16], we have shown how the analytical solution can be derived, here we give the final result for $(u(s), w(s)), s \in [0, S], u_0 \neq u_1$:

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

where the hyperbolic cosine, sine, and tangent 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

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

Fig. 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 half-ellipse through (u_0, w_0) and (u_1, w_1) with equation

$$\left(\frac{u - u_0 + w_0 \sinh r_0/\rho^2}{w_0 \cosh r_0/\rho^2}\right)^2 + \left(\frac{w}{w_0 \cosh r_0}\right)^2 = 1, \tag{10}$$

with $w > 0$. For $\rho = 1$ paths are circles. The center lies on the u-axis. Hence, when traveling over the path, in the end, the horizontal axis (corresponding to the image itself) is approached perpendicularly. In other words, if we are close to the image (small w), then panning is not effective for optimal paths. The equation of the half-ellipse that contains all points at the same distance S of a start point (u_0, w_0) is

$$\left(\frac{u - u_0}{w_0 \sinh(\rho S/2)/\rho^2}\right)^2 + \left(\frac{w - w_0 \cosh(\rho S)}{w_0 \sinh(\rho S)}\right)^2 = 1, \tag{11}$$

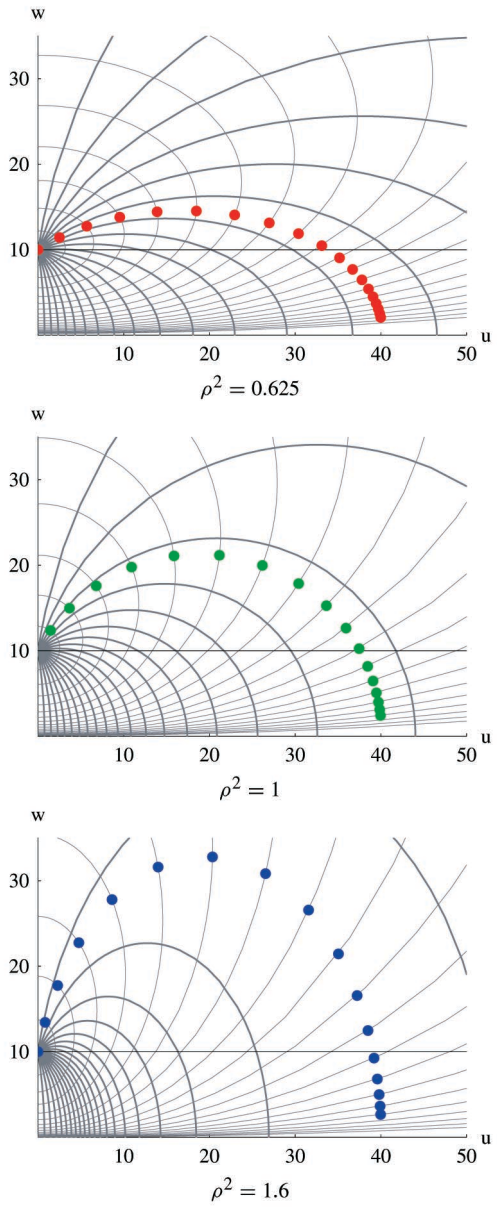

Fig. 5. Paths and iso-distance curves in (u, w) space.

with $w > 0$. Generalizing, replacement of $u - u_0$ by $|\mathbf{c} - \mathbf{c}_0|$ gives a half-ellipsoid in $(\mathbf{c}, w)$ space that denotes all points at the same distance.

Fig. 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

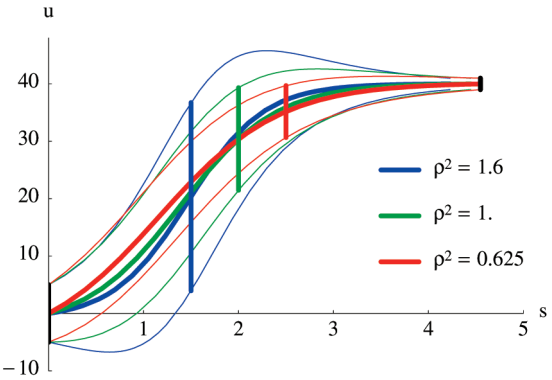

Fig. 6. u and w as a function of s.

distance $s = 4$. For a zoom out with a factor $r = w_1/w_0$, 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 10 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 stateless 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 negligible when compared to the rendering of the frame.

4.3 User Experiments

We have done a first user experiment to obtain insight into 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 [5]. This image is 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; second, a tour around 10 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

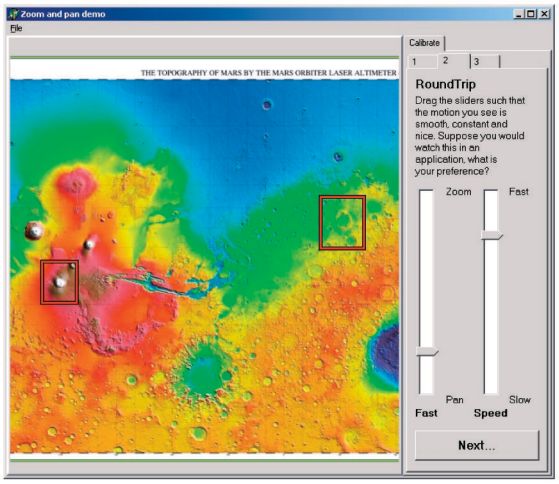

Fig. 7. Interface for user experiment.

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. Fig. 7 shows the interface. An overview of Mars is shown here and the two locations for the alternating zoom-pan animations are shown. Fig. 8 shows the average optimal animation selected by the users as a sequence of views, taken at intervals of 0.5 seconds.

Many users commented on the importance of cognitive aspects. 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.

Fig. 9 shows the parameter settings that were selected by our users. It shows that preferences for ρ and V were uncorrelated; hence, these seem to be independent dimensions, and that the conditions (two versus 10 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 confidence intervals are shown for a confidence level of 99 percent. The value of 1.42 for ρ is (statistically significant) smaller than 1.565, the value which we derived from using the root mean squared average velocity. The average value found is close to $\rho = \sqrt{2}$. It goes too far to extrapolate from our limited empirical result to such a precise suggestion for an optimal value, especially because we have no model to explain this. It is interesting though that the associated maximum width for the zoom-pan-zoom path is simply $|u_1 - u_0|$, without extra constants. For the optimal path, we find that for $w_0 = w_1$ and $\rho = \sqrt{2}$ the maximum width also has a simple form, i.e.,

$$\sqrt{(u_1 - u_0)^2 + w_0^2}.$$

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.

5 Applications

Animation from one view to another is just one, but a fundamental aspect of interactive viewing in general. Using a geometric analogy, if a view corresponds to point, an optimal animation corresponds to a line. If we know how to produce lines and how to interpolate between points, we can use this also for other purposes. In this section, we present two other applications of the model. We have not yet performed user tests to validate whether these results are indeed significant improvements in real-world cases, however, the initial results are promising.

5.1 Automatic Zooming

Igarashi and Hinckley [8] have introduced speed-dependent automatic zooming for browsing large documents. In our terminology, they choose w such that $|\dot{u}|/w$ is constant. When applied straightforwardly, this gives strongly varying values for w when $|\dot{u}|$ changes abruptly such that the aim of a constant perceived velocity is not achieved. Hence, a number of heuristics are used to bound $\dot{w}$.

The model presented here can also be used for this purpose. Suppose that the user constantly specifies values for u and/or $\dot{u}$, for instance, by dragging a scrollbar or an arrow or rectangle in a navigation window, and that, for $\dot{u} = 0$, the user wants to view his image with the minimal width w_M. Suppose further that the current view is $\mathbf{p}_0$ and that we can derive a target view $\mathbf{p}_T$ from u and/or $\dot{u}$. Let $\mathbf{p}(\mathbf{p}_0, \mathbf{p}_1, s)$ denote a point on an optimal path between two points with parameter value s and let $S(\mathbf{p}_0, \mathbf{p}_1)$ denote the corresponding distance, all according to (9). A suitable view $\mathbf{p}_\Delta$ for the next time step can then be obtained via

$$\mathbf{p}_\Delta = \mathbf{p}(\mathbf{p}_0, \mathbf{p}_T, \min(V\Delta t, S(\mathbf{p}_0, \mathbf{p}_T))).$$

In other words, we make a step along the optimal path in the direction of $\mathbf{p}_T$. The length of this step is $V\Delta t$, as long as we cannot reach the target in a single step. This assures us that the target is approached efficiently, whereas the perceived velocity is constant. The procedure is shown

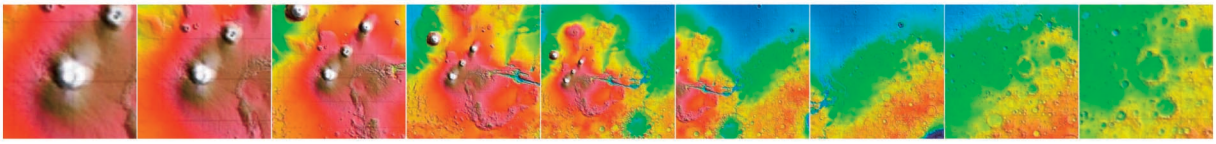

Fig. 8. Average optimal animation, time interval 0.5 seconds.

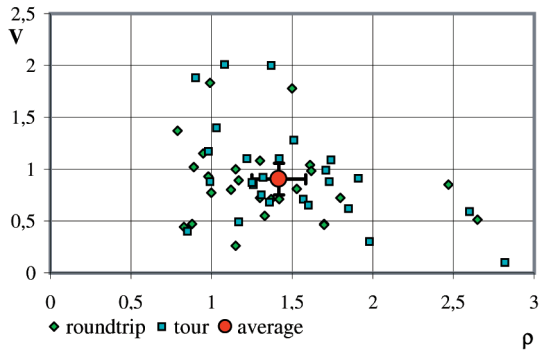

Fig. 9. Scatterplot of V and ρ values found.

schematically in Fig. 10. Blue dots denote a sequence of target views, red dots denote a sequence of derived views.

There are several possibilities for the setting of $\mathbf{p}_T$, as illustrated in Fig. 11. The blue line and dots again show target views, derived from the input of a user. From a rest state, the focus is moved with a constant speed to the right; next, the direction is reversed; and, finally, a rest state is achieved again. The value of w used here was chosen such that, in combination with $\dot{u}$, a constant perceived velocity is obtained. For a given step size $V\Delta t$ and pan velocity $\dot{u}$, the corresponding value for w follows here from

$$w_C(\dot{u}) = \max\left(w_M, \frac{\rho^2|\dot{u}|\Delta t}{2\sinh(\rho V \Delta t/2)}\right).$$

For the parts with constant speed, a good result is achieved when this input is used directly, but sudden changes in velocity cause large changes in w.

The red lines and dots show results for different strategies for $\mathbf{p}_T$. Every fifth dot is connected here with the associated target point, to show the relation more clearly. First, we can take only *positional* information into account:

$$\mathbf{p}_T = (u, w_M).$$

The width w does increase, but slowly, and, as a result, the view shown is delayed with respect to the input of the user. However, the resulting animation is smooth. Second, we can take only *velocity* information into account:

$$\mathbf{p}_T = (u_0 + \dot{u}\Delta t, w_C(\dot{u})).$$

This speed-dependent approach is similar to that of Igarashi and Hinckley. The image in the middle shows that the target w is reached quickly and smoothly. The absolute value of u is not met, but this cannot be expected here as only the velocity is used as input. Finally, we can take both *position* and *velocity* into account:

$$\mathbf{p}_T = (u, w_C(\dot{u})).$$

The position is matched better now. The value for w does overshoot to make up for the delay in the start of the sequence.

The position-based approaches suffer from lag. This is inevitable when the amount of image change is bounded. However, a smaller lag, at the expense of a higher optic flow, can be obtained by increasing V and ρ. Fig. 12 shows an example for the position controlled approach, where, for both parameters, a 30 percent higher setting was used.

Experiments with users for a number of tasks will have to reveal what their preference is. For the time being, we find it promising that, based on the model, various strategies that produce smooth results can be defined easily. Furthermore, the strategies that use position are

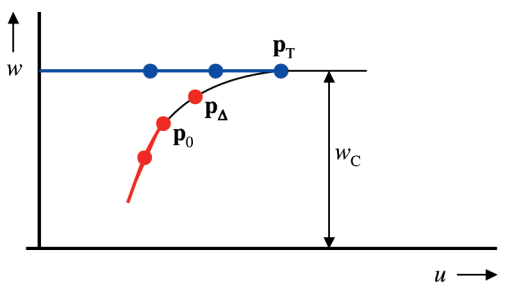

Fig. 10. Principle automatic zooming.

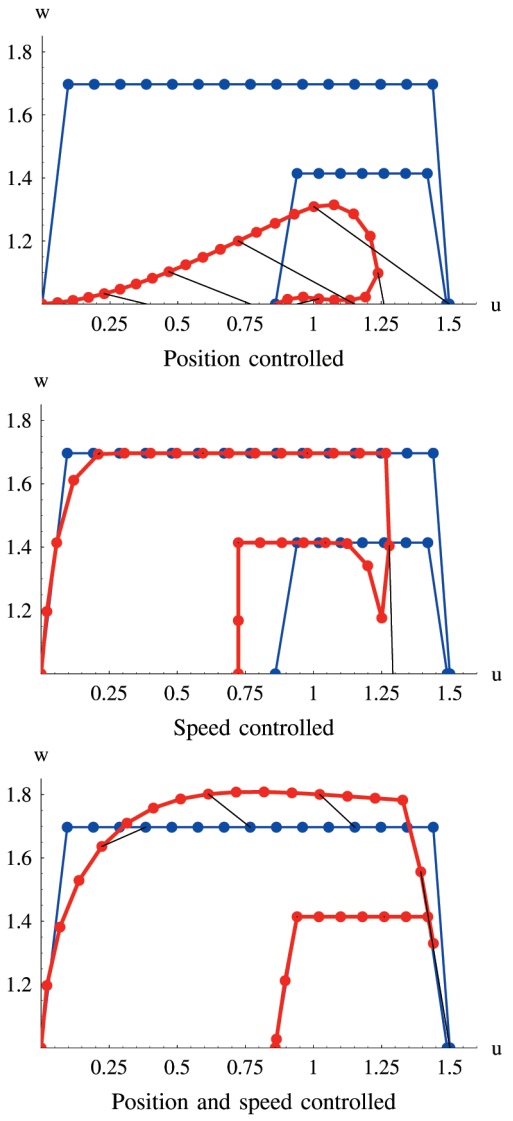

Fig. 11. Automatic zooming.

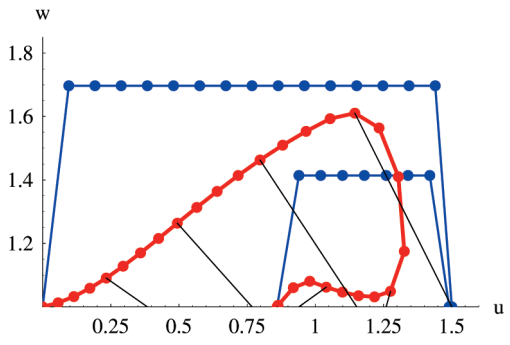

Fig. 12. Position controlled, V and ρ 30 percent increased.

compatible with the results presented before: A discontinuous change of u gives the optimal path.

5.2 Arbitrary Paths

Suppose that an arbitrary camera path, $\mathbf{p}(s^*) = (\mathbf{c}(s^*), w(s^*))$, has been defined by a designer or otherwise and that we want to achieve a constant perceived velocity along the path. This can be easily achieved via reparametrization of the path such that the path is arclength parametrized according to the metric (6). Specifically, the given path is replaced by a path $\mathbf{q}(s) = \mathbf{p}(s^*)$, where s follows from numerical integration of

$$s = \int_0^{s^*} \sqrt{\frac{\rho^2 \dot{\mathbf{c}}(x) \cdot \dot{\mathbf{c}}(x)}{w^2(x)} + \frac{\dot{w}^2(x)}{\rho^2 w^2(x)}}\, dx,$$

Instead of doing this reparametrization initially, it can also be done on the fly. Suppose that the current projection is $\mathbf{p}_0$ and that a step $S = V\Delta t$ has to be made. One approach is to evaluate the preceding integral for increasing s^* until a value is found for which $s = S$. Alternatively, the isodistance half-ellipsoid, defined in (11), can be used. The intersection of the given path with this half-ellipsoid is the next point searched for, which can be computed analytically if the path is defined by linear or quadratic segments.

6 Generic Transforms

We have shown how the model for smooth zooming and panning can be used for various applications. A next question is if the model can be extended such that also other image transforms can be dealt with, i.e., rotation and nonuniform scaling. These are more rare than translation and uniform scaling, but do occur, especially when abstract data are visualized. A good example are treemaps [9]. A treemap is a space-filling display of hierarchical data, generated by recursively subdividing rectangles in alternating directions. If one wants to zoom in smoothly on a part of the hierarchy, using the rule that on the highest level shown the first subdivision is always from left to right, both rotation and nonuniform scaling have to be dealt with. For both cases, we derive the optimal paths, which can be used as a basic building block for the applications discussed in the previous section.

6.1 Rotation

We first consider rotation, using the same line of thought as for the base case. Suppose that the view is rotated over an angle α. The projection of points (x, y) in world space to points (x', y') in image space is then given by

$$(x', y') = \left(\frac{x - c_x}{w}, \frac{y - c_y}{w}\right) \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}.$$

A measure for the average perceived velocity of the rotating, moving and scaling image is the corresponding root mean squared average velocity V_{RMS}

$$\begin{aligned} V_{\text{RMS}}^2 &= V^2 \left(\frac{1}{6}\, \dot{\alpha}^2 + \frac{1}{w^2}\, \dot{c_x}^2 + \frac{1}{w^2}\, \dot{c_y}^2 + \frac{1}{6w^2}\, \dot{w}^2\right) \\ &= V^2 \left(\frac{1}{6}\, \dot{\alpha}^2 + \frac{1}{w^2}\, \dot{u}^2 + \frac{1}{6w^2}\, \dot{w}^2\right). \end{aligned}$$

We see that the impact of rotation is limited. The effect of $\dot{\alpha}$ is independent on the current scale w. Also, there is no interaction between rotation and position, hence an optimal path will be along a line through $\mathbf{c}_0$ and $\mathbf{c}_1$ and we use u as a parameter along this line again.

The value of V_{RMS} leads to the following metric on (α, u, w) space

$$ds^2 = \frac{1}{\mu^2}\, d\alpha^2 + \frac{\rho^2}{w^2}\, du^2 + \frac{1}{\rho^2 w^2}\, dw^2,$$

where μ is an extra parameter that denotes the visual impact of rotation, just like ρ denotes the impact of scaling. The unit of μ is radians/second, a large value for μ indicates that rotation has little visual impact. We have not done a user study yet to find an optimal value. The value for V_{RMS} suggests using $\mu = \sqrt{6}$, which we found to be acceptable in practice.

To find an optimal path, we derive a system of differential equations from this metric, i.e., the arc length parametrization constraint and the geodesic equations

$$\begin{aligned} &\dot{\alpha}^2/\mu^2 + \rho^2 \dot{u}^2/w^2 + \dot{w}^2/\rho^2 w^2 = 1, \\ &\ddot{\alpha} = 0, \\ &\ddot{u} - 2\dot{u}\dot{w}/w = 0, \text{ and} \\ &\ddot{w} + \rho^4 \dot{u}^2/w - \dot{w}^2/w = 0. \end{aligned}$$

The last two equations are the same as for the base case. The optimal path for an animation from (α_0, u_0, w_0) to (α_1, u_1, w_1) is hence also similar as for the base case:

$$\begin{aligned} \alpha(s) &= (\alpha_1 - \alpha_0)s/S + \alpha_0 \\ u(s) &= \frac{w_0}{\rho^2} \cosh r_0 \tanh\left((r_1 - r_0)s/S + r_0\right) \\ &\quad - \frac{w_0}{\rho^2} \sinh r_0 + u_0, \\ w(s) &= w_0 \cosh r_0 / \cosh\left((r_1 - r_0)s/S + r_0\right), \\ S &= \sqrt{(\alpha_1 - \alpha_0)^2/\mu^2 + (r_1 - r_0)^2/\rho^2}, \\ r_i &= \ln(-b_i + \sqrt{b_i^2 + 1}),\ i = 0, 1, \text{ and} \\ b_i &= \frac{w_1^2 - w_0^2 + (-1)^i \rho^4 (u_1 - u_0)^2}{2 w_i \rho^2 (u_1 - u_0)},\ i = 0, 1. \end{aligned}$$

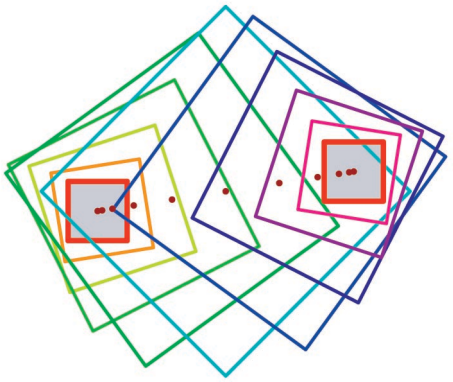

Fig. 13. Example rotation.

The angle α is just linearly interpolated from α_0 to α_1, the distance S increases to compensate for the extra rotation to be done. Fig. 13 shows an example. Start and end views are shown as gray boxes in world space, the views in between are shown as rectangles, whose centers are shown as dots.

6.2 Nonuniform Scaling

We next consider nonuniform scaling. Suppose that the width of the view is specified independently for the x and y direction. The projection of points (x, y) in world space to points (x', y') in image space is then given by

$$(x', y') = \left(\frac{x - c_x}{w_x}, \frac{y - c_y}{w_y}\right)\begin{pmatrix}\cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha\end{pmatrix}.$$

Note that, with this definition (translation, scaling, followed by rotation), the axes remain orthogonal. A measure for the average perceived velocity is

$$V_{\mathrm{RMS}}^2 = V^2\left(\frac{1}{6}\,\dot{\alpha}^2 + \frac{12\dot{c_x}^2 + \dot{w_x}^2}{12w_x^2} + \frac{12\dot{c_y}^2 + \dot{w_y}^2}{12w_y^2}\right).$$

We see that this measure falls apart into three parts: one for rotation and one per axis for translation and scaling. The latter parts each have the same structure as the measure derived for the base case. Because of this separation, we can no longer make the assumption that the optimal path is along a straight line between the two center points.

From the value of V_{RMS}, we derive the following metric on $(\alpha, c_x, c_y, w_x, w_y)$ space:

$$ds^2 = \frac{d\alpha^2}{\mu^2} + \frac{\rho^2 dc_x^2}{w_x^2} + \frac{dw_x^2}{2\,\rho^2 w_x^2} + \frac{\rho^2 dc_y^2}{w_y^2} + \frac{dw_y^2}{2\,\rho^2 w_y^2}.$$

This metric is chosen such that it leads to the same optimal path as presented before for a purely symmetric case ($w_{xi} = w_{yi}$ and $c_{xi} = c_{yi}, i = 0, 1$).

The associated differential equations, i.e., the arc length parametrization constraint and the geodesic equations, are

$$\frac{\dot{\alpha}^2}{\mu^2} + \frac{\rho^2\dot{c_x}^2}{w_x^2} + \frac{\dot{w_x}^2}{2\,\rho^2 w_x^2} + \frac{\rho^2\dot{c_y}^2}{w_y^2} + \frac{\dot{w_y}^2}{2\,\rho^2 w_y^2} = 1,$$

$$\ddot{\alpha} = 0,$$

$$\ddot{c}_z - 2\dot{c}_z\dot{w}_z/w_z = 0, \text{ and}$$

$$\ddot{w}_z + 2\,\rho^4\dot{c}_z^2/w_z - \dot{w}_z^2/w_z = 0,$$

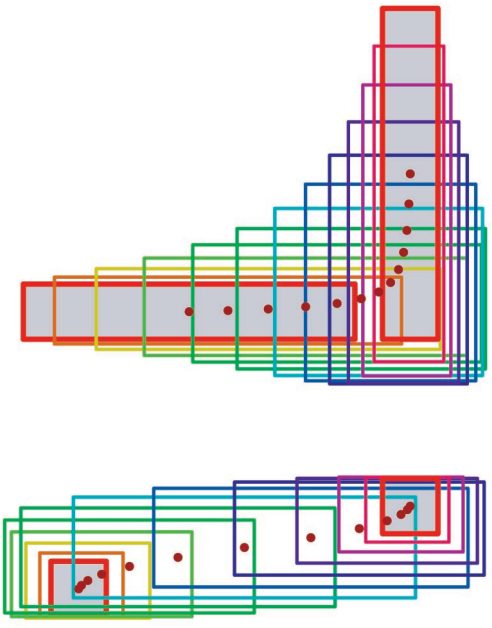

Fig. 14. Examples of nonuniform scaling.

where z denotes, here and in the next equations, x or y. The optimal path for an animation from $(\alpha_0, \mathbf{c}_0, w_{x0}, w_{y0})$ to $(\alpha_1, \mathbf{c}_1, w_{x1}, w_{y1})$ is now

$$\begin{aligned}
\alpha(s) &= (\alpha_1 - \alpha_0)s/S + \alpha_0 \\
c_z(s) &= \frac{w_{z0}}{\sqrt{2}\rho^2}\cosh r_{z0}\tanh\left((r_{z1} - r_{z0})s/S + r_{z0}\right) \\
&\quad - \frac{w_{z0}}{\sqrt{2}\rho^2}\sinh r_{z0} + c_{z0}, \\
w_z(s) &= w_{z0}\cosh r_{z0}/\cosh\left((r_{z1} - r_{z0})s/S + r_{z0}\right), \\
S &= \sqrt{\frac{(\alpha_1 - \alpha_0)^2}{\mu^2} + \frac{(r_{x1} - r_{x0})^2}{2\,\rho^2} + \frac{(r_{y1} - r_{y0})^2}{2\,\rho^2}}, \\
r_{zi} &= \ln(-b_{zi} + \sqrt{b_{zi}^2 + 1}),\ i = 0, 1, \text{ and} \\
b_{zi} &= \frac{w_{z1}^2 - w_{z0}^2 + 2(-1)^i\rho^4(c_{z1} - c_{z0})^2}{2\sqrt{2}\,w_{zi}\,\rho^2(c_{z1} - c_{z0})},\ i = 0, 1.
\end{aligned}$$

Fig. 14 shows examples of nonuniform scaling. The top image shows a transition from a horizontal strip to a vertical one. At first, a horizontal motion dominates. Meanwhile, the strip distorts into a square and, next, to a vertical strip. In the last part of the animation, vertical motion dominates. For such situations, a smooth and pleasing result is obtained.

One would hope and expect that this general model gives a good result for arbitrary transitions. Unfortunately, this is not the case; the bottom image shows a counter example. Here, a simple transition from one square to another has to be made. The square is distorted into a broad rectangle here because, in the horizontal direction, a much larger distance has to be crossed than in the vertical direction. When viewed on the screen, this looks like a cartoon effect. This particular path is indeed the most efficient path between the two views in four-dimensional (c_x, c_y, w_x, w_y) space, but, for viewing, for instance, a cartographic map, one would prefer that the views in

between are also square. A remedy could be to add the additional constraint that the aspect ratio has to remain constant or, more generally, has to increase or decrease monotonically, but we have not worked this out further yet.

7 Conclusions

We have shown how to generate smooth animations when viewing 2D images. The core of the model is the definition of a metric on the impact of zooming, panning, and rotation, which we used to produce smooth camera paths for various applications. The original impetus for this work was the visualization of call graphs via a hierarchical matrix visualization [15]. 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.

A number of aspects deserve further study. The extensions presented in Sections 5 and 6 should be evaluated with rigorous user-studies. We have not yet derived a method for nonuniform scaling with a constraint on the change of the aspect-ratio. A possible variation is to use a varying velocity such as acceleration at the beginning and deceleration at the end of the animation, thereby mimicking real world camera movements. User-studies are required to find out if this is preferable to a constant velocity.

We aimed at perceptually pleasing results, however, cognitive aspects also play an important role for this kind of animation. 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 parameters to his preferences.

Another interesting area for future research concerns the applicability of this model for 3D navigation. We obtained a very first result, in the sense that this work provides a theoretical confirmation of some of the empirical results provided by Mackinlay et al. [12]. A generic solution for optimal paths in 3D would be very useful for, for instance, architectural walk-throughs. However, in 3D navigation, the complexity of the optic flow pattern is much higher than for 2D image viewing and depends strongly on the scene and the current point of view; hence, we cannot expect that a closed form solution can be derived.

Appendix

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

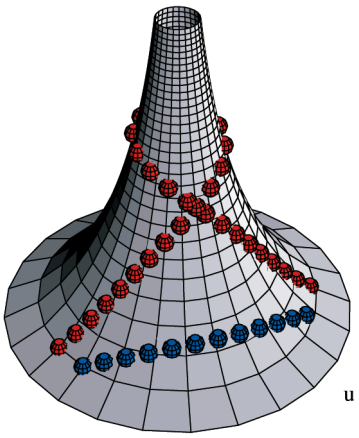

Fig. 15. u, w space depicted as a geometric surface.

distort the 2D images shown in Fig. 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.

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.,

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

Fig. 15 shows a surface that satisfies this constraint:

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

For $\rho = 1$, this is a well-known surface in geometry: a pseudosphere. Similar to a sphere, the Gaussian curvature is constant, but here it is negative instead of positive.

Back to our context, the horizontal circles are lines with constant w, the lines toward the top are lines with constant u. In other words, panning is mapped to rotating around the central axis. Panning over large distances corresponds

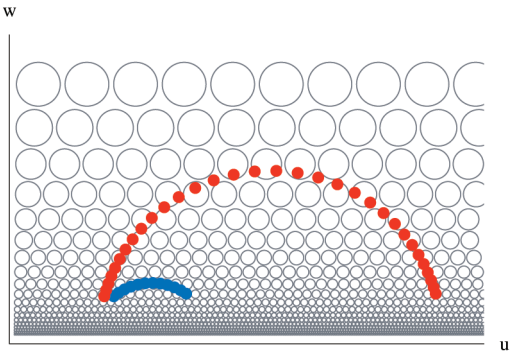

Fig. 16. Trajectories in u, w space.

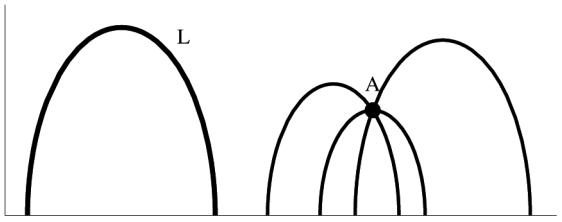

Fig. 17. Hyperbolic zoom-pan space: Multiple lines through A.

to rotating multiple times around the central axis, each point on the surface is multivalued. 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. Fig. 16 shows the same trajectories in u, w space.

We can illustrate that we are dealing with non-Euclidean space also in another way. Euclid's fifth axiom states that, given a line L and a point A not on this line, there exists a unique line through A that does not intersect L. In our situation, however, if we replace lines by geodesics, there exist an infinite number of lines that do not intersect the given line (Fig. 17). The latter is characteristic for *hyperbolic geometry*, discovered independently by Bolyai and Lobachevsky in the early 19th century. One model for hyperbolic geometry is Poincaré's Disc model, well-known from the art of M.C. Escher and applied in Information Visualization for fish-eye views [10]. Another model, also from Poincaré, is the Half-Plane model, which is the same as the model used here (for $\rho = 1$).

ACKNOWLEDGMENTS

The authors are grateful to Luc Florack, Frank van Ham, and Jean-Bernard Martens (all TU Eindhoven) for their inspiration and support

REFERENCES

[1] L. Auslander, *Differential Geometry,* p. 165. Harper & Row, 1967.
[2] B.B. Bederson, J. Meyer, and L. Good, "Jazz: An Extensible Zoomable User Interface Graphics Toolkit in Java," *Proc. UIST 2000, ACM Symp. User Interface Software and Technology,* pp. 171-180, 2000.
[3] B.B. Bederson, L. Stead, and J.D. Hollan, "Pad++: Advances in Multiscale Interfaces," *Proc. CHI '94 Human Factors in Computing Systems,* pp. 315-316, 1994.
[4] D. Eberly, "A Differential Geometric Approach to Anisotropic Diffusion," *Geometry-Driven Diffusion in Computer Vision,* B.M. ter Haar Romeny, ed., pp. 371-392, Dordrecht: KAP, 1994.
[5] D.E. Smith et al., "Mars Orbiter Laser Altimeter (MOLA): Experiment Summary after the First Year of Global Mapping of Mars," *J. Geophysical Research,* vol. 106, no. 23, pp. 689-722, 2001. See also http://ltpwww.gsfc.nasa.gov/tharsis/global.map.html.
[6] G.F. Furnas, "Generalised Fisheye Views," *Proc. CHI '86 Human Factors in Computing Systems,* pp. 16-23, 1986.
[7] G.W. Furnas and B.B. Bederson, "Space-Scale Diagrams: Understanding Multiscale Interfaces," *Proc. CHI '95 Human Factors in Computing Systems,* pp. 234-241, 1995.
[8] T. Igarashi and K. Hinckley, "Speed-Dependent Automatic Zooming for Browsing Large Documents," *CHI Letters,* vol. 2, no. 2, pp. 139-148, 2000. *Proc. UIST 2000, ACM Symp. User Interface Software and Technology.*
[9] B. Johnson and B. Shneiderman, "Treemaps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures," *Proc. Second Int'l IEEE Visualization Conf.,* pp. 284-291, Oct. 1991.
[10] J. Lamping, R. Rao, and P. Pirolli, "A Focus+Content Technique Based on Hyperbolic Geometry for Viewing Large Hierarchies," *Proc. ACM SIGCHI Conf. Human Factors in Computing Systems (CHI '95),* pp. 401-408, May 1995.
[11] M. Lipschutz, *Differential Geometry,* p. 234. McGraw-Hill, 1969.
[12] J.D. Mackinlay, S.K. Card, and G.G. Robertson, "Rapid Controlled Movement through a Virtual 3D Workspace," *Computer Graphics,* vol. 24, no. 4, pp. 171-176, 1990. Proc. SIGGRAPH '90.
[13] B. Shneiderman, "The Eyes Have It: A Task by Data Type Taxonomy of Information Visualizations," *Proc. IEEE Symp. Visual Languages '96,* pp. 336-343, 1996.
[14] D.J. Struik, *Lectures in Classical Differential Geometry,* p. 142. Addison-Wesley, 1950.
[15] F. van Ham, "Using Multi-Level Call Matrices in Large Software Projects," *Proc. IEEE Symp. Information Visualization 2003 (INFOVIS2003),* T. Munzner and S. North, eds., pp. 227-232, 2003.
[16] J.J. van Wijk and W.A.A. Nuij, "Smooth and Efficient Zooming and Panning," *Proc. IEEE Symp. Information Visualization 2003 (INFOVIS2003),* T. Munzner and S. North, eds., pp. 15-22, 2003.

Jarke J. van Wijk received the MSc degree in industrial design engineering in 1982 and the PhD degree in computer science in 1986, both with honors. He is a full professor of visualization at the Technische Universiteit Eindhoven. His main research interests are information visualization and flow visualization, both with a focus on the development of new visual representations.

Wim Nuij received the MSc degree in mathematics fromt the Leiden University in 1964. He is an assistant professor at the Technische Universiteit Eindhoven. In the 1990s, he extended his interests to computer graphics.