A Framework for Non-Realistic Projections

by

Jonathan Levene

B.S., Massachusetts Institute of Technology (1997)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1998

Author .....
Department of Electrical Engineering and Computer Science
May 8, 1998

Certified by .....
Julie Dorsey
Associate Professor
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

A Framework for Non-Realistic Projections

by

Jonathan Levene

Submitted to the Department of Electrical Engineering and Computer Science
on May 8, 1998, in partial fulfillment of the
requirements for the degree of
Master of Engineering

Abstract

Over the last thirty years, most research efforts in computer graphics have been directed towards producing photorealistic images. Photorealism, however, exhibits some very real shortcomings in practical and artistic settings. Non-photorealistic renderers (NPRs) offer new solutions by abandoning the accurate modeling of optics in order to achieve more expressive results. Previous NPR systems have focused on simulating traditional media by displaying geometric results in a non-realistic fashion. By abandoning photorealism, however, NPR systems also have the option of performing projection, in addition to lighting and visibility resolution, non-realistically. Projection techniques are particularly useful as a means of controlling how information is presented to a viewer, expressing the various ways in which the shape of objects, and the spatial relations between them, can be represented in pictures.

We describe a framework for interactively computing non-realistic projections from 3D world space to 2D screen space. The framework provides a means of (1) using curved projection surfaces; (2) controlling the degree to which orthogonals converge to or diverge from a vanishing point in the image; (3) controlling the behavior of orthogonals as they converge or diverge; and (4) projecting different objects independently and compositing their images together. We demonstrate our approach with a variety of expressive projections applied to complex models.

Thesis Supervisor: Julie Dorsey
Title: Associate Professor

Acknowledgements

The author would like to thank the MIT Computer Graphics Group for providing machines, office space, and words of encouragement throughout the course of this work. Special thanks go to Prof. Julie Dorsey, for her insight and suggestions; Sami Shalabi and Michael Capps, for warmly and very patiently sharing their incredible knowledge of systems and networking; Hans Pedersen and Justin Legakis, for taking the time to help with theoretical and algorithmic roadblocks; Max Chen, for invaluable help with Alias modeling; and finally Luke Weisman, whose inspiration and support was fundamental to initially getting this work off the ground.

Contents

List of Figures

Chapter 1

Introduction

Computer graphics, like many other fields of research in computer science, has been marked by great improvements over the last thirty years. The vast majority of these developments have been directed towards one goal: the production of *photorealistic* images (that is, images indistinguishable from photographs of actual objects and scenes). The extent to which researchers have succeeded in this aim can be seen by viewing modern feature films and TV commercials, where it is often impossible to distinguish the "real thing" from pictures of a simulation.

Why has research within the graphics community been biased towards realism? This is due to the fact that computer graphics is rooted in both art and science, two disciplines which have historically espoused realism. Since the Renaissance, western art has focused on representing the physical appearance of the world around us as accurately as possible (a tendency reinforced by the invention of photography). Computer graphics was heavily influenced by this focus. In addition, though, computer graphics developed as a science that could model material objects and light in the environment and predict the way in which they interact. Such predictive models became increasingly useful for certain applications, such as automobile or architectural design.

The nature and successes of realism have led many people to claim it as more "objectively correct" than other forms of representation, thus making it more desirable. When examined from practical and artistic points of view, however, photorealism

exhibits some very real shortcomings.

1.1 The practical shortcomings of photorealism

In many applications, photorealistic images are sought to give "true and natural" representations of a scene. In contrast to this view, more and more philosophers and art historians in recent years have noted the rather unnatural nature of photographs. Gombrich [8], for example, shows how abstract a photograph really is through exposing the many artifices it employs: the micro-instant frozen forever, the limited angle of view, and the seeming arbitrariness of photographic processing.

If we look at the photograph in terms of its practical role as a means of visual communication, it is clearly not an optimal representation. Strothotte et al. [16] describe how photorealistic images fail to convey implicit information to a viewer. They write:

> More subtle and at the same time even less considered in the literature is the question of how a viewer is influenced by a graphical presentation. Graphics do not only convey information encoded intentionally or explicitly in an image or in the underlying model, but easily go well beyond. This situation is analogous to textual communication where a reader may "read between the lines": A rendered image can convey a great deal of information that is not represented explicitly in the input model of today's renderers.

As an example, they cite the domain of architecture where CAD models and conventional renderers are often used to produce an image of a design. One problem with such images is that they often do not reflect the degree of development of the design (i.e. whether the design is preliminary, of production quality, or somewhere in between). They confidently convey the message that the object being displayed is complete and well thought out, even if it is only a first draft. A second problem is that architects often feel that a rendered image of an object appears "stale" compared

to the sense of vitality offered by hand-drawn graphics. Thus, rendered images are sometimes traced over by hand with pencil and paper, a tedious and unnecessary task [16].

Winkenbach and Salesin [19] explain how photorealistic images often fail to communicate complex information in a comprehensible and effective manner. In order to communicate truly complex information effectively, some form of visual abstraction is required. Hand-drawn illustrations, they explain, convey information better by omitting extraneous detail, by focusing attention on relevant features, by clarifying and simplifying shapes, or by exposing parts that are hidden. Illustrations also provide a more natural vehicle for conveying information at different levels of detail. As an example, they explain how medical texts almost always employ hand-drawn illustrations in place of (or in addition to) photographs, since they allow tiny and hidden structures to be much better described. In addition, most assembly, maintenance, and repair manuals of mechanical hardware employ illustrations rather than photographs because of their clarity. At Boeing, even when CAD databases of airplane parts exist, all high-quality manuals are still illustrated by hand in order to provide more effective diagrams than can be achieved with either photorealistic rendering or simple hidden line drawings [19]. Lansdown and Schofield [11] also point out the shortcomings of photographs by asking, "How much use is a photograph to mechanics when they already have the real thing in front of them?"

1.2 The artistic shortcomings of photorealism

From the perspective of an artist, an image has many other uses than simply imitating nature. Gombrich [8] has suggested that the goals of art are many-fold: exploring patterns, expression and abstraction, symbolism, magic, the mental world, and the perceptual world. A more abstract form of image generation (*non-photorealistic* image generation) is a necessary tool for such art.

As an example, consider painting a natural scene. As Meier [14] explains, the power of painting lies in its expressiveness: one can

- reduce a subject to its essence by not depicting every detail. The reduction of detail can direct a viewer's eye to an area of interest and can also allow the viewer to "complete the picture and share in the interpretative process."

- exaggerate the effect of light to create a "wide tonal range that creates richness and drama".

- use different brush strokes to define the character of a surface and how light is reflected from it. One can imply smoothness or softness with well-blended strokes, or use direct, unblended strokes to imply stronger lighting and more pronounced surface texture.

- add rhythm to a painting by varying the definition of edges. One can paint edges that are distinct in one place and "lose themselves in another", or even let brush strokes cross edge boundaries altogether.

- imply depth by varying brush size. Larger, smoother brush strokes tend to recede in depth, while small, textured strokes better depict foreground detail.

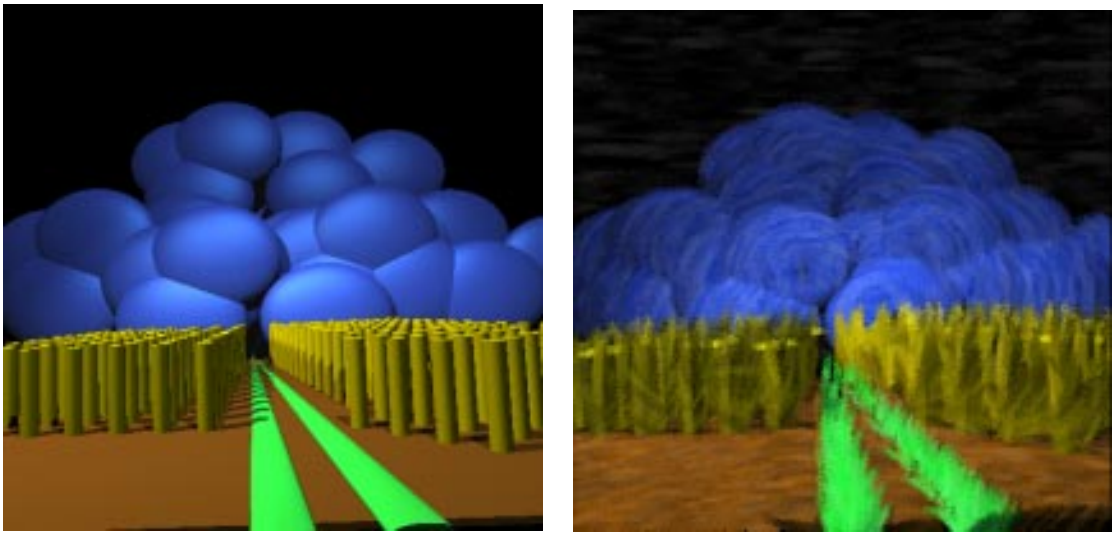
A visual example of the expressiveness of these non-photorealistic effects is shown in Figure 1-1.

1.3 Previous non-photorealistic geometry-based systems

Despite the attention that photorealistic techniques have received within the computer graphics community, a number of non-photorealistic systems have been built in recent years. These systems can be classified into two broad categories, depending on their input: *geometry-based systems*, which take 3D scene descriptions as input, and *image-based systems*, which take images as input.

Both types of systems have advantages and disadvantages. Geometry-based systems (which we restrict ourselves to here) have full access to 3D geometry, lighting,

(a)



(b)

Figure 1-1: Comparing photorealistic and non-photorealistic images. In (a), Edvard Munch's *The Scream* was modeled using 3D geometry and was then rendered twice. The left-most image was generated from the author's photorealistic renderer (a recursive ray tracer) while the right-most image was generated from the author's non-photorealistic renderer (a recursive ray tracer fashioned after [14]). In (b), a model of Van Gogh's *Threatening Skies* was processed by the same two renderers to produce the images on the left and right, respectively.

and viewing information. This allows such systems to produce illustrations whose strokes can convey (1) the tone and texture of surfaces in the scene, and (2) the 3D form of these surfaces. The disadvantage of these systems is that the geometric models they require, along with the reflectance properties of each surface in the model, are often non-trivial to create and modify. Image-based systems, however, greatly reduce the required complexity of their input. They can accept a physical photograph, computer-generated image, or even a 2D visualization of non-physical data as input. The disadvantage with these systems is that they have no knowledge of the underlying geometry or the viewing transformation behind the images they are rendering. This makes it very difficult, for example, to render different objects in the image in different styles.

Previous non-photorealistic geometry-based systems were architected after the sketch-renderer presented in [16]. This architecture incorporates the notions of *transmitted* and *transputed* information, both of which are used as inputs to the renderer. Transmitted information is that which is encoded in the 3D geometry, and describes the elements of the scene and the way in which they are lit. Transputed information, however, is the major artistic contribution to the final image. It affects the viewer's perception by altering the way in which geometric and lighting information are presented.

The architecture is shown graphically in Figure 1-2. Here, transmitted information is designed to be independent of transputed information. A modeler uses modeling software to create the geometric models and lights used as input to the renderer. A designer can then alter non-photorealistic parameters in the renderer to control the transputed content. Distinguishing transmitted from transputed information is important because, traditionally, transputed information was controlled by re-modeling the scene to achieve a desired look. This, clearly, was a long and tedious process.

In addition to sharing a common architecture, most non-photorealistic geometry-based systems have focused on simulating traditional media. In recent years, various systems have been implemented that simulate pen-and-ink illustration [3, 4, 15, 16, 19, 20, 12], paint [9, 14], watercolor [2], and combinations thereof [11]. It is important

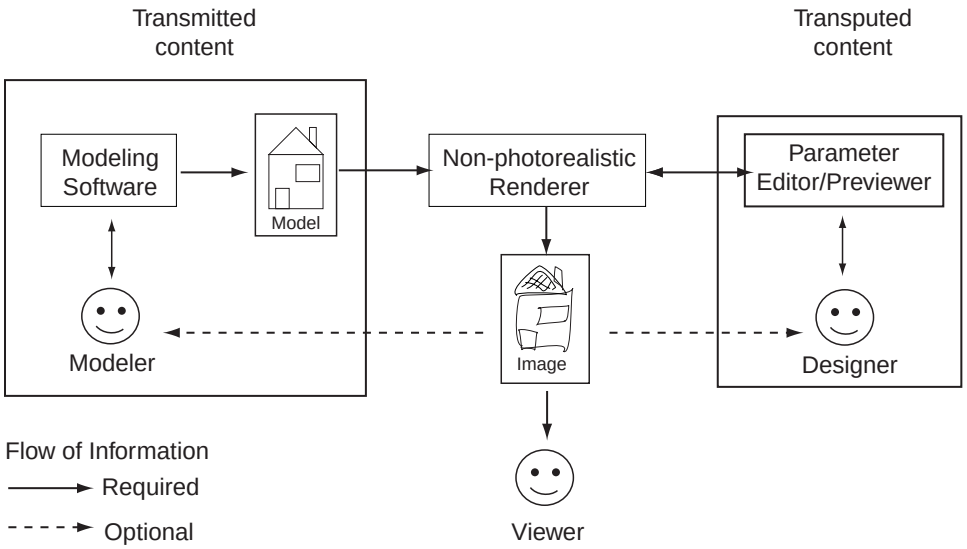


Figure 1-2: Architecture of a non-photorealistic renderer. (Diagram reproduced from [16])

to note that these systems were not created to replace artist with machine; rather, they were created to automate the application of strokes in artwork, leaving "artistic decisions" (i.e. transputed information) to be supplied by the user. Once viewed this way, non-photorealistic renderers (NPRs) become tools for artists and designers to quickly and more easily visualize their ideas.

1.4 Non-photorealistic lighting, projection, and visibility

As its name suggests, NPR is *not photorealistic*: it abandons the accurate modeling of optics in order to achieve more expressive results.

The classical rendering pipeline, a sequence of steps that scan-converting renderers follow in converting 3D geometry to a 2D image, indicates how photorealistic renderers achieve a degree of optical accuracy. Figure 1-3 shows one simple version of the pipeline based on [7]. As the figure indicates, the pipeline is a five stage process:

1. **Lighting**: once scene geometry is input into a scan-converting renderer, the geometry is lit at each vertex according to a local lighting model.

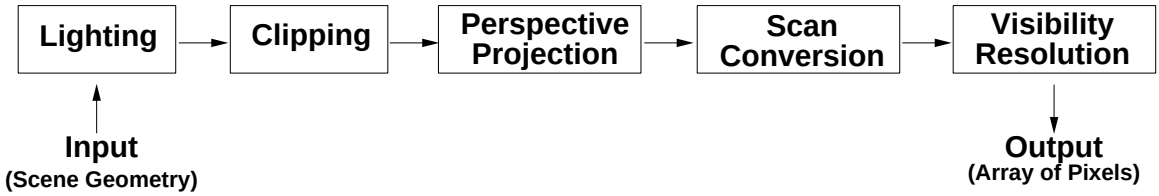


Figure 1-3: The classical rendering pipeline.

2. **Clipping**: portions of the scene that lie outside a 3D view volume are removed or "clipped".

3. **Perspective Projection**: the scene is projected onto a 2D image plane. A portion of the image plane is then mapped to "screen space" (defined by a viewport).

4. **Scan Conversion**: each object in screen space is diced into horizontal spans, one per viewport raster. Each object's depth and color are then interpolated along its boundaries.

5. **Visibility Resolution**: the renderer determines which object is visible at each pixel, and outputs the final image.

Although today's renderers often vary the order of the steps in the pipeline and the way in which they are carried out, certain invariants remain. Every renderer must light its scene, project it onto a 2D surface, resolve visibility, and display its geometric results. Each of these four steps may be done realistically or not, independently of one another. In simulating traditional media, previous NPR systems perform lighting, projection, and visibility realistically, while displaying geometric results non-realistically by applying a medium onto a canvas in a sophisticated manner. The techniques used in applying a medium range from using texture mapping for each stroke, to using a time-varying model of how the medium interacts with the fibers on the canvas itself.

Traditional media simulation is, however, only one of many possible NPR styles. Since NPR is based on the idea of abandoning optics, we also have the option of performing projection, in addition to lighting and visibility resolution, non-realistically.

Projections are of particular importance in that they provide a simple and powerful means of representing 3D space, and the objects it contains, in an image. To our knowledge, investigating the use of projections in representing 3D space has attracted surprisingly little attention. In this thesis, we present a framework for non-realistic projections (alternatives to planar geometric perspective) that serve as new and powerful spatial representations.

Chapter 2

Non-realistic projections

In this chapter, we define non-realistic projections and describe their many uses. First, however, we describe and categorize previous non-perspective rendering systems.

2.1 Previous non-perspective rendering systems

Previous non-perspective systems fall into one of two categories. First, some have addressed the problem of correcting distortions in images. Zorin et al. [22] addressed distortions inherent in perspective transformations. Such distortions were corrected by employing perceptually preferable viewing transformations, each of which could be decomposed into a perspective or parallel projection followed by a planar transformation. Two other systems addressed distortions resulting from the projection of an image onto a viewing surface. Max [13] addressed 180° projection onto a domed theater, while Dorsey et al. [5] addressed projection onto a planar viewing surface from an angle. Both of these systems were designed to account for the resulting distortion by pre-distorting their output with a non-linear projection.

Second, other systems have addressed the problem of visualizing 3D space in an image. Wood et al. [21] built a system to render a single background image, called a multi-perspective panorama, which incorporated multiple views of a 3D environment as seen from along a given camera path. When viewed through a small moving window, the panorama produced the illusion of 3D motion. Inakage [10] built a sys-

tem to explore the use of non-linear perspective projections in achieving particular expressive effects. The major contribution of this work was to focus on how information is presented to a viewer via three new techniques: (1) curvilinear projection; (2) inverted perspective projection; and (3) 3D warping. The system presented in this thesis extends Inakage's work by increasing the range of alternative projections according to a unifying framework.

2.2 The properties of perspective

In defining non-realistic projections, it will first be of use to describe the properties of planar geometric perspective. Perspective is the means by which cameras, and indeed the human visual system, capture the 3D world into a 2D image.

Perspective projection can be characterized by the convergence of rays of light (called projectors) onto a finite point. Figure 2-1 shows how projectors converge in both cameras and in computer simulations when a perspective projection is performed. In a camera, incoming rays of light which have been either emitted or reflected from nearby objects are restricted to pass through a tiny aperture, forming a cone of light. Once past the aperture, the light rays fall onto a picture plane, forming the projection image. The resulting image in this case is inverted and backwards, as is shown in Figure 2-1. Computer-based photorealistic renderers solve this problem by moving the projection plane in front of the aperture, and renaming the aperture as "the eye". The projection in this case is taken as the intersection of the projection plane and all incoming rays that converge at the eye.

Whether perspective projection is performed by camera or by computer, incoming rays converge to a point. This property is the cause of the two most salient features of perspective:

- **Diminution of size with distance**: The image of an object decreases in size as the object moves further away from the eye.

- **Vanishing points**: Orthogonals (lines which are parallel to each other but

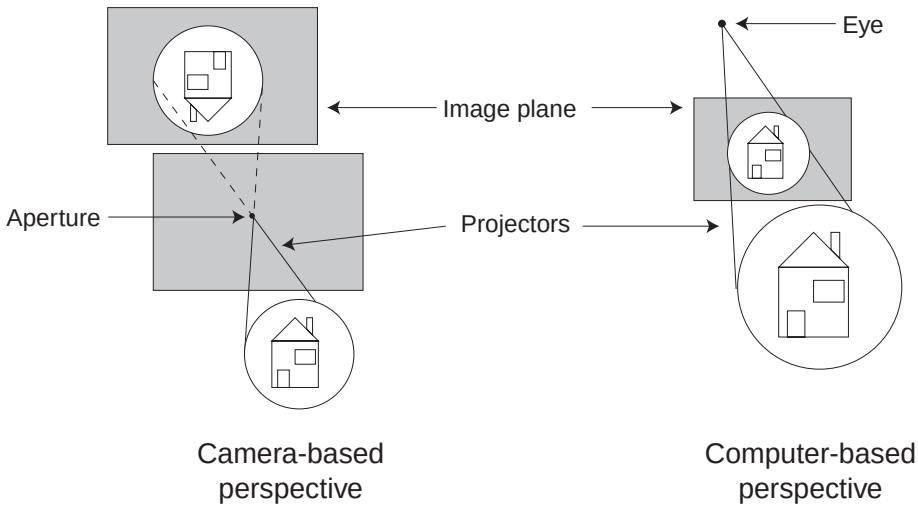


Figure 2-1: Camera-based and computer-based perspective.

orthogonal to the projection plane) converge to a "vanishing point" in the image.

These features can clearly be seen in the left image of Figure 1-1 (b). Here, the wheat stalks that are further away appear smaller than those in front, indicating the diminution of size with distance. In addition, the two parallel rows of grass converge off in the distance, indicating the existence of a vanishing point.

2.3 Defining non-realistic projections

In this section, we define non-realistic projections and explain some of their advantages over conventional perspective projection.

We informally characterize a projection as being parameterized by:

1. The shape of the projection surface.
2. The degree to which orthogonals converge to or diverge from a vanishing point in the image.
3. The behavior of orthogonals as they converge or diverge.
4. The subset of objects in the scene that are operated on.

According to this definition, planar geometric perspective is characterized by (1) a planar projection surface; (2) orthogonals that converge to a vanishing point; (3) orthogonals that converge in straight lines; and (4) the entire scene being viewed. We characterize non-realistic projections as differing from perspective in one or more of these parameters. For example, in *oblique* projection, orthogonals neither converge nor diverge but are represented as straight parallel lines running obliquely across the image. This forces object size to remain constant with distance. In *inverse perspective*, orthogonals diverge from vanishing points in straight lines, forcing object size to increase with distance [1].

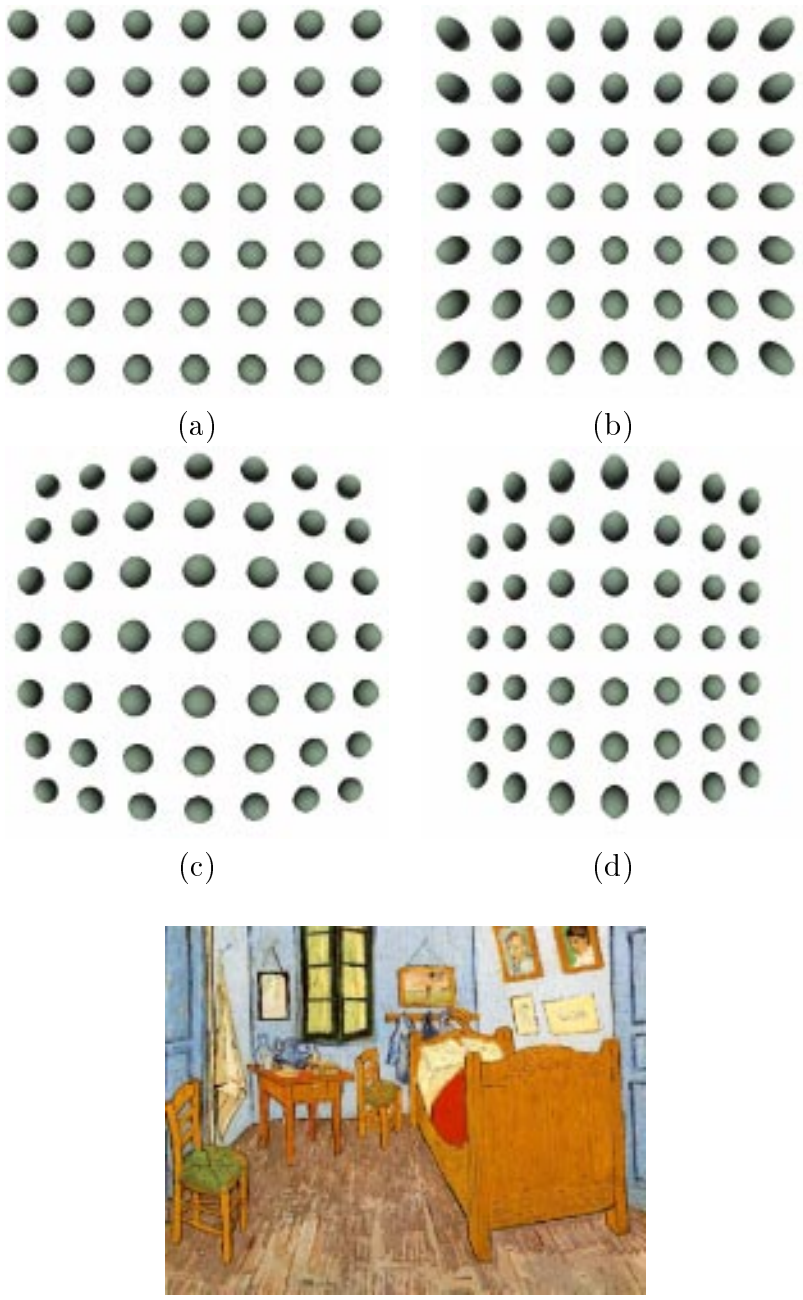
The advantage of using non-realistic projections defined in this way is that one can control how information is presented to the viewer by minimizing deformed and ambiguous shape representations, by controlling how object detail varies with distance, and by using multiple viewpoints.

2.3.1 Shape representations

One important role of any projection is to convey a representation of each object's 3D shape. While planar geometric perspective has many strengths, such as its usefulness in conveying depth, it often presents a distorted and ambiguous description of shape in three different ways. As we shall see, controlling (1) the shape of the projection surface, and (2) the degree to which orthogonals converge or diverge in the image offers some solutions to these problems.

First, using a planar projection surface often causes an object's shape to be both distorted and ambiguous. As is well known, objects represented in perspective suffer major deformations when viewed from a wide angle [6]. In these situations, objects that lie in the periphery of a wide field of view may be stretched outward - as shown in Figure 2-2b. Using a curved projection surface, as shown in Figures 2-2c and 2-2d, is an effective means of reducing wide angle distortion and ensuring a degree of shape constancy in an object's image. In addition, curved projection surfaces also have

[1] For a more thorough explanation of these projections, see the book by Willats [18].



(a) (b) (c) (d) (e)

Figure 2-2: Two advantages of using curved projection surfaces. Curved projection surfaces are useful in minimizing wide-angle distortion. Here, a grid of spheres has been rendered using four different projections: in narrow-angle (45°) planar perspective (a), in wide-angle (90°) planar perspective (b), in wide-angle spherical perspective (c), and in wide-angle cylindrical perspective (d). In addition, curved projection surfaces are also useful for viewing objects in the periphery of an image. In (e) is shown Van Gogh's *Bedroom of Arles*.

the property of slightly rotating objects lying in the periphery of the image towards the viewer. This is useful in exposing the shape of objects which would normally be viewed edge-on, so that more of their side view is visible. As an example, consider Van Gogh's *Bedroom of Arles* shown in Figure 2-2e, which appears as though the bedroom has been projected onto a cylindrical surface. If we look at the bed, the result is that we can see more details of its side than we would in planar geometric perspective. With a planar projection surface, we would be viewing the bed from nearly edge-on, resulting in an ambiguous representation of its true shape.

A second problem with perspective is that the illusion of depth it provides also distorts an object's shape representation. In perspective, objects can be made to appear distant by tilting or rotating them away from the projection plane. In the resulting image, sizes, shapes, spatial distances, and angles must be distorted simply to convey depth [1]. If, in the eye of a designer or engineer, conveying a scene's depth is secondary in importance to conveying the shape and location of its objects, an oblique projection might be a more direct and immediate approach. An example of this is shown in Figure 2-3. Here, the oblique image has many advantages over the perspective one. The size, orientation, and location of each person at the table is clear, and it is also clear that the table is square and not rectangular.

Third, as the psychologist Arnheim argues, perspective is well suited for representing interior spaces but offers ambiguous representations of exterior views of shapes [1]. This is true for two reasons. In perspective, each location on the projection plane corresponds to multiple parts of an object or to multiple objects. Also, because orthogonals converge in the image, foreground tends to hide background and side faces of objects tend to be hidden. These problems can be seen in the perspective-drawn altarpiece of Figure 2-4. The inverse perspective drawing, however, does not suffer from these same problems. In the image of the altarpiece, each location on the projection plane corresponds to a single point on either a front or side face. In addition, since the more distant parts of the altarpiece are larger than nearby ones, its side faces are spread out and their details exposed.

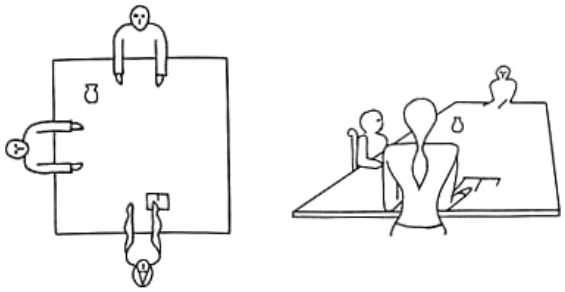


Figure 2-3: Two ways of representing persons sitting around a table. The left image uses an oblique-like projection, while the right image uses a perspective-like projection. (Illustration taken from [1])

Figure 2-4: The effect of converging and diverging orthogonals. Left: a detail from a Spanish altarpiece of the fourteenth century, drawn with diverging orthogonals. Right: the same subject drawn with converging orthogonals. (Illustration taken from [1])

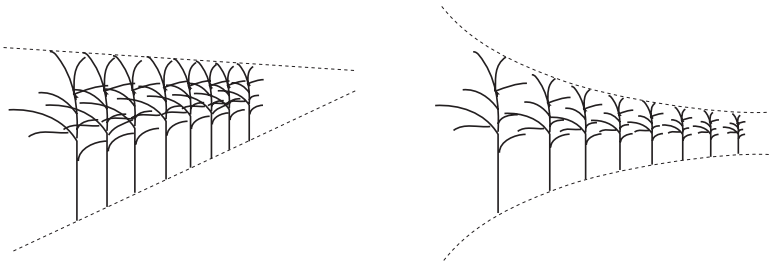


Figure 2-5: The effect of linear and curved orthogonal convergence. Two illustrations of a tree line extending into the distance, both drawn with converging orthogonals. Left: convergence occurs linearly. Right: orthogonals curve as they converge.

2.3.2 Varying object detail with distance

In perspective, the fact that orthogonals converge in straight lines can sometimes obscure the detail of distant objects. For example, in a row of objects extending into the distance, as shown in Figure 2-5, distant ones are obscured by those nearby because of the linear convergence of their orthogonals. Forcing orthogonals to curve to a vanishing point reveals more detail of distant objects as they shrink in size by smearing their location across the canvas. This effectively reduces the degree to which they are occluded.

2.3.3 Multiple viewpoints

In perspective, only a single point of view of the entire scene is presented. Using multiple viewpoints in an image can reduce occlusion, allowing the viewer to be more "mobile" by seeing around objects and perceiving faces that would ordinarily be seen edge-on. An example of this technique is beautifully illustrated in Cézanne's *Still Life with Fruit Basket*, shown in Figure 2-6. Notice how the left and right sides of the table do not line up, indicating that each is viewed from a different height. Both the fruit basket, and the large jar immediately to its left, are also displayed from more of a top view than would be physically possible, which has the advantage of revealing their contents.

Multiple viewpoints can be used by employing a set of non-realistic projections, in which each operates on a subset of the scene. This enables objects to be projected independently and composited into a final image.

Having defined non-realistic projections and described their uses, we now present the framework for our system, followed by a demonstration of its capabilities.

Figure 2-6: Illustrating the use of multiple viewpoints. Left: Cézanne's *Still Life with Fruit Basket*. Right: an illustration isolating parts of the scene drawn from different viewpoints.

Chapter 3

Framework

Our system is designed to allow a user to interactively define multiple projections, assign a subset of the scene to each, operate on each subset, and finally composite the scene back together to form the final image. Each projection is defined by visually setting each of its four parameters (as described above). In this section, we present the underlying projection method of the system, followed by details of how each parameter is implemented.

3.1 Underlying projection method

Our process of projecting a subset of a scene involves four steps:

1. **Discretization**: the scene is divided into a set of triangles according to a user-specified level of coarseness.

2. **Lighting**: each vertex of each triangle is shaded in world space.

3. **Transformation**: each vertex is re-expressed in orthographic eye space and then transformed by altering its x and y coordinates depending on its radial distance from the eye.

4. **Orthographic projection**: each vertex is then projected by discarding its z value. This z value is stored for use in resolving visibility during rendering.

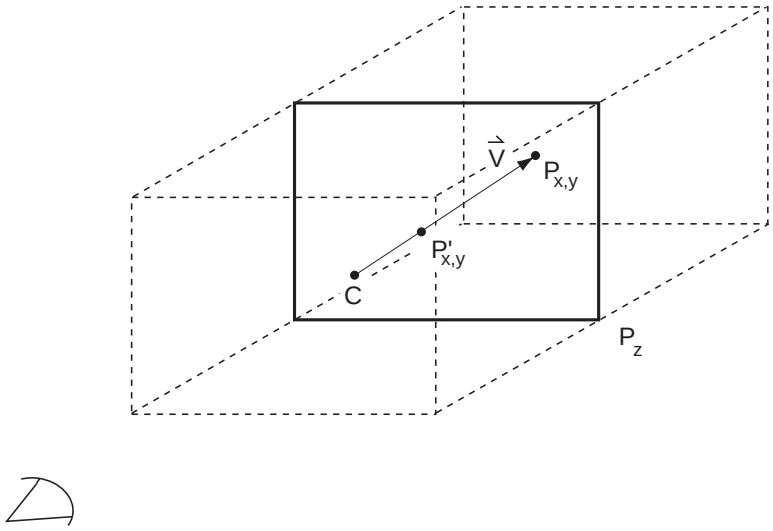


Figure 3-1: The transformation step. P is scaled in 2D by $f(P)$, relative to C.

The characteristics of each non-realistic projection are embedded in the transformation step. The transformation, depicted in Figure 3-1, simply scales an eye-space point P in two dimensions by a function f relative to some center of scaling C. In formal terms, given point P in eye space, the transformation P' is given by

$$\begin{aligned} P' \quad = \quad & (C_x + [f(P) \cdot \vec{V}_x], \\ & C_y + [f(P) \cdot \vec{V}_y], \\ & P_z) \end{aligned} \tag{3.1}$$

where

- the point C, called the *center of scaling*, is a 2D point at the same eye-space depth as P.
- the vector $\vec{V} = (P_x - C_x, P_y - C_y)$ is a 2D vector describing P relative to C.
- the function $f : \mathbf{R}^3 \longmapsto \mathbf{R}$, called the *size-with-distance* function, determines the factor to scale the x and y components of P, relative to C.

As we will explain, the point C is used to control the location of the vanishing points in the image, while f is used to control the degree to which orthogonals converge

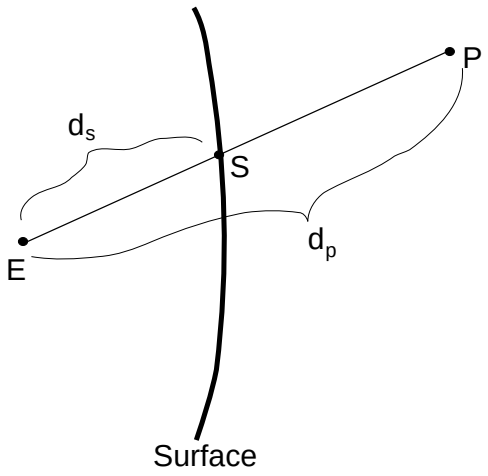


Figure 3-2: Calculating $f(P)$, the size-with-distance function.

or diverge. We define f to be

$$f(P) = \left(\frac{d_p}{d_s}\right)^n \tag{3.2}$$

where

- E is the eye point.
- S is the projection of P onto a viewing surface.
- $d_p = |P - E|$ is the radial distance of P from the eye.
- $d_s = |S - E|$ is the radial distance of S from the eye.
- n is a scalar that determines the degree to which orthogonals diverge or converge (see Figure 3-2).

3.2 Projection surface

In our system, the shape of the projection surface determines d_s, influencing the degree to which a point is scaled in eye space. The surface is represented as a non-parametric cubic Bézier patch of the form:

$$z = f(\alpha, \beta)$$

where

- $\alpha = \tan^{-1}(x)$ and $\beta = \tan^{-1}(y)$ are measures of a direction vector's angular displacement in x and y respectively. To convert a 3D vector to $\alpha - \beta$ space, we first scale it until it intersects the plane $z = 1$. Its $\alpha - \beta$ representation is then $(\tan^{-1}(x), \tan^{-1}(y))$. Converting back to Cartesian space would then yield $(x, y, z) = (\tan(\alpha), \tan(\beta), 1)$.

- z is the depth of the surface point along the $\hat{z}$ direction in eye-space. The surface point itself is calculated by finding where the direction vector (α, β) intersects the plane $z = f(\alpha, \beta)$.

The sixteen control points of the Bézier patch are evenly distributed in what we call $\alpha - \beta$ space, and the user may interactively alter each point's corresponding depth.

With this representation, we simplify our surface in that we do not allow for self intersection or self occlusion. However, the advantage is that calculating S (the projection of point P onto the surface) requires only that we find the $\alpha - \beta$ representation for the vector $\overrightarrow{P - E}$, and then look up the point on the surface lying in that direction.

3.3 Converging and diverging orthogonals

The degree to which orthogonals converge to or diverge from a vanishing point is controlled by the exponent n in Equation 3.2. The sign of n controls whether orthogonals converge or diverge, while the magnitude controls how quickly convergence or divergence occurs. For example, if $n = -1$, $f \propto \frac{1}{d_p}$, and we have a perspective-like projection. If $n = 2$, $f \propto {d_p}^2$, and we have an "accelerated" inverse perspective-like projection (where divergence occurs twice as quickly). If $n = 0$, we have an orthographic projection.

In our system, a value for n can be set for a particular depth in eye space. We implement this feature by having the user set four values of n at equally spaced z intervals, and then interpolating values between them using a cubic Bézier curve.

By allowing n to vary smoothly through the depth of the scene, our system can interpolate between perspective, orthographic, and inverse perspective projections by eye-space depth.

As we have described it, our framework does not support oblique projections, where orthogonals remain parallel yet run across the image surface at an oblique angle. This is because each non-realistic projection is implemented with an underlying orthographic projection, where projectors hit the image plane at right angles. For oblique projections, we would require the view volume to be skewed, so that projectors still remain parallel yet intersect the image plane at an oblique angle.

To enable oblique projections, as is shown in Figure 3-3, we allow the user to interactively skew the view volume in eye space by setting the x and y coordinates of a *skew point*, K. After a skew point is defined, each point P is skewed into point P_{skew}, before being transformed, according to:

$$\begin{aligned} P_{skew} &= (P_x - [K_x \cdot (\frac{P_z}{K_z})], \\ &\quad P_y - [K_y \cdot (\frac{P_z}{K_z})], \\ &\quad P_z) \end{aligned}$$

After skewing, each point P_{skew} is transformed into P'_{skew} and orthographically projected, as described earlier.

3.4 Manipulating vanishing points

As mentioned earlier, the center of scaling C in Equation 3.1 controls the location of vanishing points in the image. For all points at the same depth in eye space, the vanishing points are offset from their location by the vector $\vec{C}$.

In the case of a perspective projection (where $n = -1$), if C is constant throughout the depth of the scene then point P will shrink towards C as it gets further away from the eye. If $C = (0, 0)$, each vanishing point will remain untouched. If C is a non-zero vector, each vanishing point will be offset.

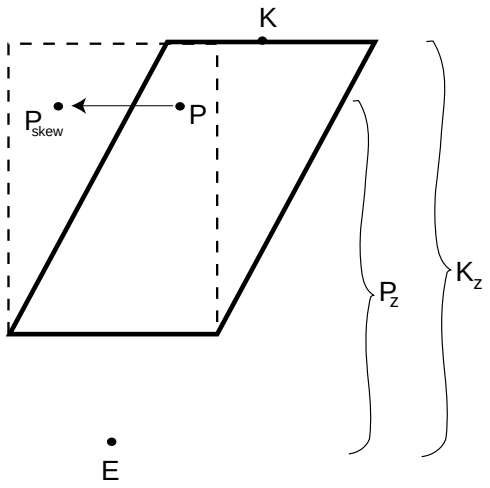


Figure 3-3: Skewing the orthographic view volume to implement oblique projections.

In our system, the user can set a location for C for a particular depth in eye space. This is done by setting four values of C at equally spaced z intervals, and then interpolating between them using another cubic Bézier curve. Shifting the location of the curve, while keeping it straight, results in a shift of all vanishing points. Bending the curve forces C to change by depth, resulting in orthogonals that curve to a vanishing point.

3.5 Combining multiple projections

Because each non-realistic projection may operate on a subset of the scene, multiple projections may be combined in our system. Naively assembling the transformation of each subset before performing orthographic projection, however, drastically changes the composition of the scene as the position, scale, and relative orientation of objects is altered. What we desire is a method of constraining the position, scale, and orientation of objects while branding their images with the characteristics of their corresponding projection.

3.5.1 Inertial fitting

We solve this problem by merging multiple projections together with a 2D inertial fitting technique. If a scene is partitioned into m projections, one is arbitrarily assigned the "default" projection (which we call p_d). The image of each projection p_i $(1 \leq i \leq m-1)$ is then merged into that of p_d according to the following two steps:

1. The objects assigned to p_i are **duplicated**. One copy is then transformed by p_i, the other by p_d. This yields two images of the original set of objects.

2. The first image is **fitted** to the second so that the position, orientation, and size of the images match.

To match the position of the images, the 2D centroid of each image is calculated. The first image is then translated so that its centroid matches that of the second.

To match the orientation of the images, the axes of minimum and maximum inertia are aligned. The axes for each image are simply the eigenvectors of its corresponding 2D inertial matrix:

$$\begin{bmatrix} \iiint x^2\,dx\,dy\,dz & \iiint xy\,dx\,dy\,dz \\ \\ \iiint xy\,dx\,dy\,dz & \iiint y^2\,dx\,dy\,dz \end{bmatrix}$$

Aligning the pairs of axes can be viewed as simply a change of orthobasis. A 2D rotation matrix R, which performs the change of basis, is given by:

$$R = T \cdot S^{-1}$$

where

- S and T are matrices whose columns are the inertial axes of the source and target images, respectively.

Finally, the image sizes are matched by scaling the x and y components of the source image so that its 2D bounding circle matches that of the target image.

3.5.2 Shortcomings

This fitting technique is one possible solution to the problem of combining multiple projections, although it is a rather simplistic one. One shortcoming is that visibility is not correctly resolved. Objects which should be occluded in the scene may emerge to the forefront during merging. A second shortcoming is that the success of this technique depends on the orientation of the camera relative to the object being projected. If one of the two resulting images of the object extends nearly equally in the x and y dimensions (which can result from positioning the camera appropriately), no unique axes of inertia may exist. Thus, the matching of orientation may fail.

To deal with this second shortcoming, one might be tempted to extend the technique into three dimensions: one could use 3D axes of inertia to match orientation and use bounding spheres to match scale. Our framework for projections, however, precludes this approach. This is because both images of an object extend equally in the z dimension, since the z coordinate of each point in eye space remains untouched after projection. Thus, the axes of inertia, in addition to the bounding spheres, will always be biased in the z-direction.

As we will discuss later, combining multiple projections remains a topic for further research.

Chapter 4

The interactive system

4.1 The interface

We provide an editor that allows a user to interactively define multiple projections, assign a subset of objects to each, transform each subset, and finally render the result. Figure 4-1 shows a snapshot of our system in use.

The *transform window* (left) allows the user to orthographically view the transformed scene from arbitrary angles.

On the right, three windows define the current projection. The *function window* (top right) allows the user to interpolate between perspective, orthographic, and inverse perspective projection through the depth of the scene, altering the degree to which orthogonals converge or diverge in the image. This is done by dragging any of the four control points of the Bézier curve defining how n changes with depth.

The *view volume window* (middle right) allows the user to shape the projection surface, control how orthogonals converge or diverge, and skew the view volume. The projection surface is shaped by dragging any of the sixteen control points on the Bézier patch, altering their distance from the eye. Making orthogonals curve is accomplished by dragging any of the four control points of the Bézier curve defining how C changes with depth. Skewing the view volume is accomplished by dragging the skew point K lying on the far plane of the view volume.

The *visualization window* (bottom right) displays the subset of the scene assigned

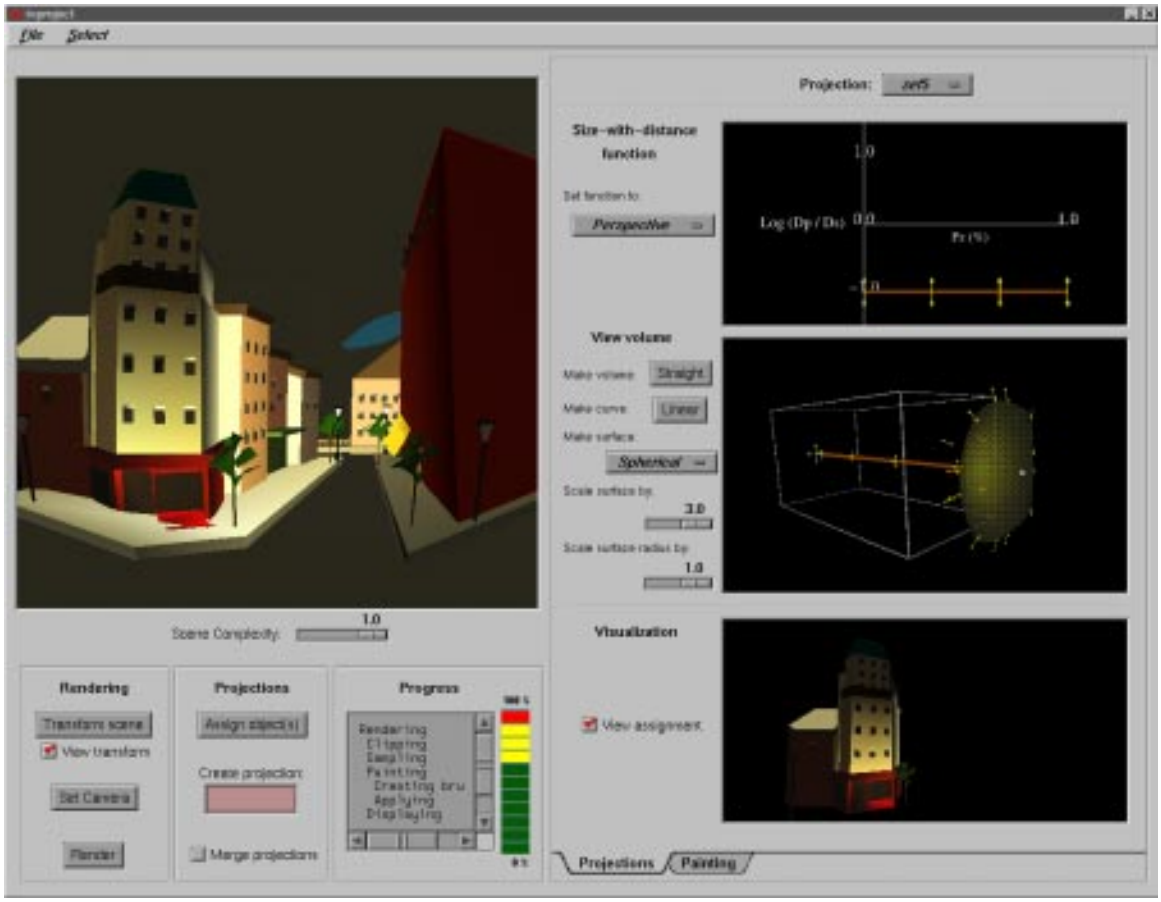


Figure 4-1: A snapshot of our system in use.

to the current projection.

4.2 Defining projections

Figures 4-2 and 4-3 illustrate the result of interactively varying three different parameters in defining projections. Figure 4-2 illustrates the effect of varying the shape of the Bézier curve for n. Along the x axis, the two "near" control points vary from -1 to 1, setting the value of n in the near half of the depth of the scene. This effects the size, shape, and composition of the wheat field in the foreground. Along the y axis, the two "far" control points vary from -1 to 1, setting the value of n in the far half of the depth of the scene. This effects the size, shape, and composition of the clouds in the background. The rows of grass stretching into the scene illustrates how n is smoothly interpolated through intermediate depths.

Figure 4-3 illustrates the effect of varying both the shape of the Bézier surface and the Bézier curve for C. Along the x axis, the shape of the Bézier surface varies from concave to convex, relative to the eye. Here, we can see how concave projection surfaces enlarge objects near the focal point while shrinking objects in the periphery of the image. Convex projection surfaces shrink objects near the focal point while enlarging objects in the periphery. These effects are most prominent in the clouds of the background. Along the y axis, the two "far" control points of the Bézier curve for C vary from the left to the right, setting the location of C in the far half of the depth of the scene. This forces the both the wheat stalks and the rows of grass to curve to a vanishing point as depth increases.

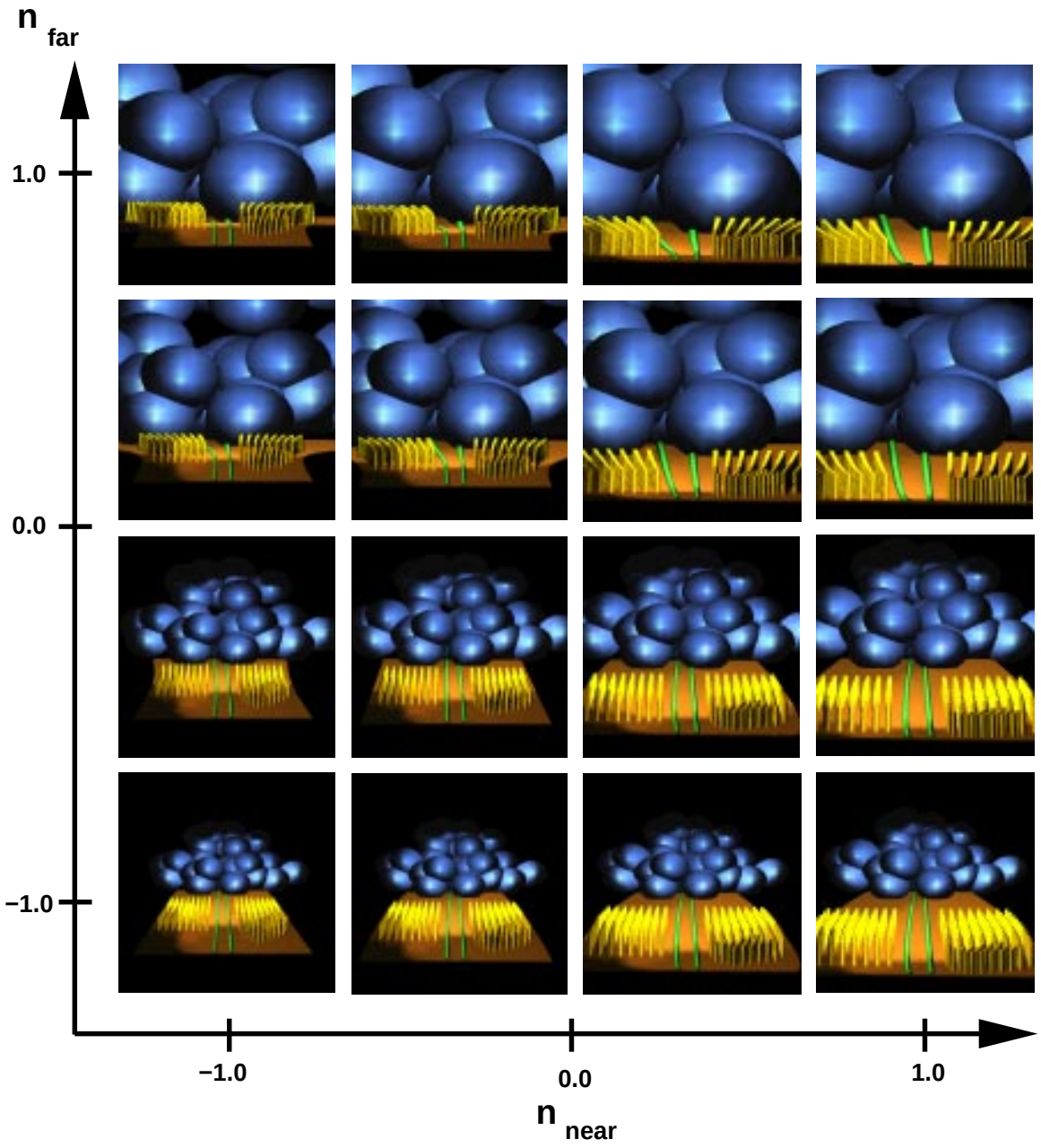


Figure 4-2: The effect of varying n with depth. A test scene has been rendered sixteen times while varying both n in the near half of the scene and n in the far half.

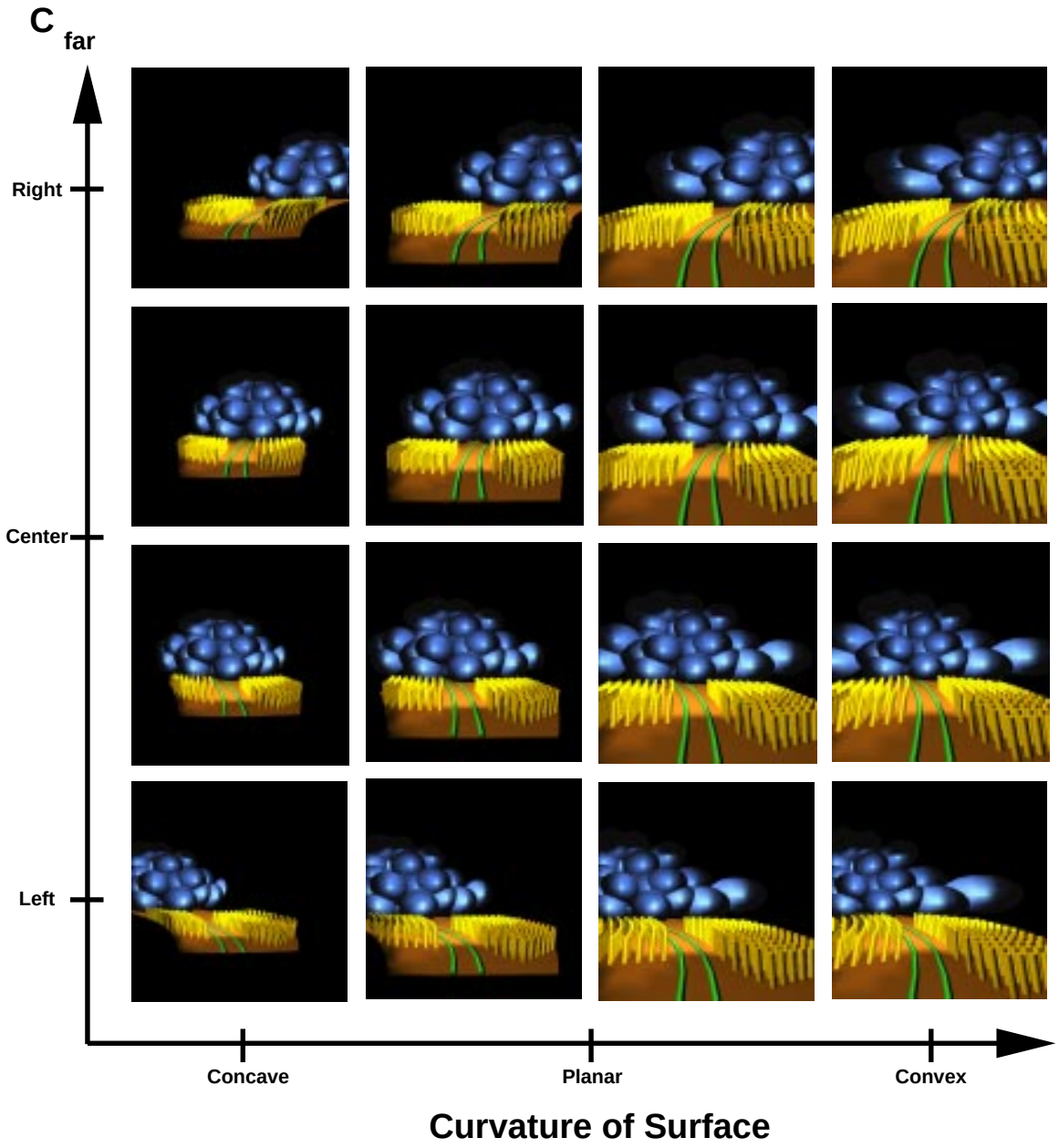


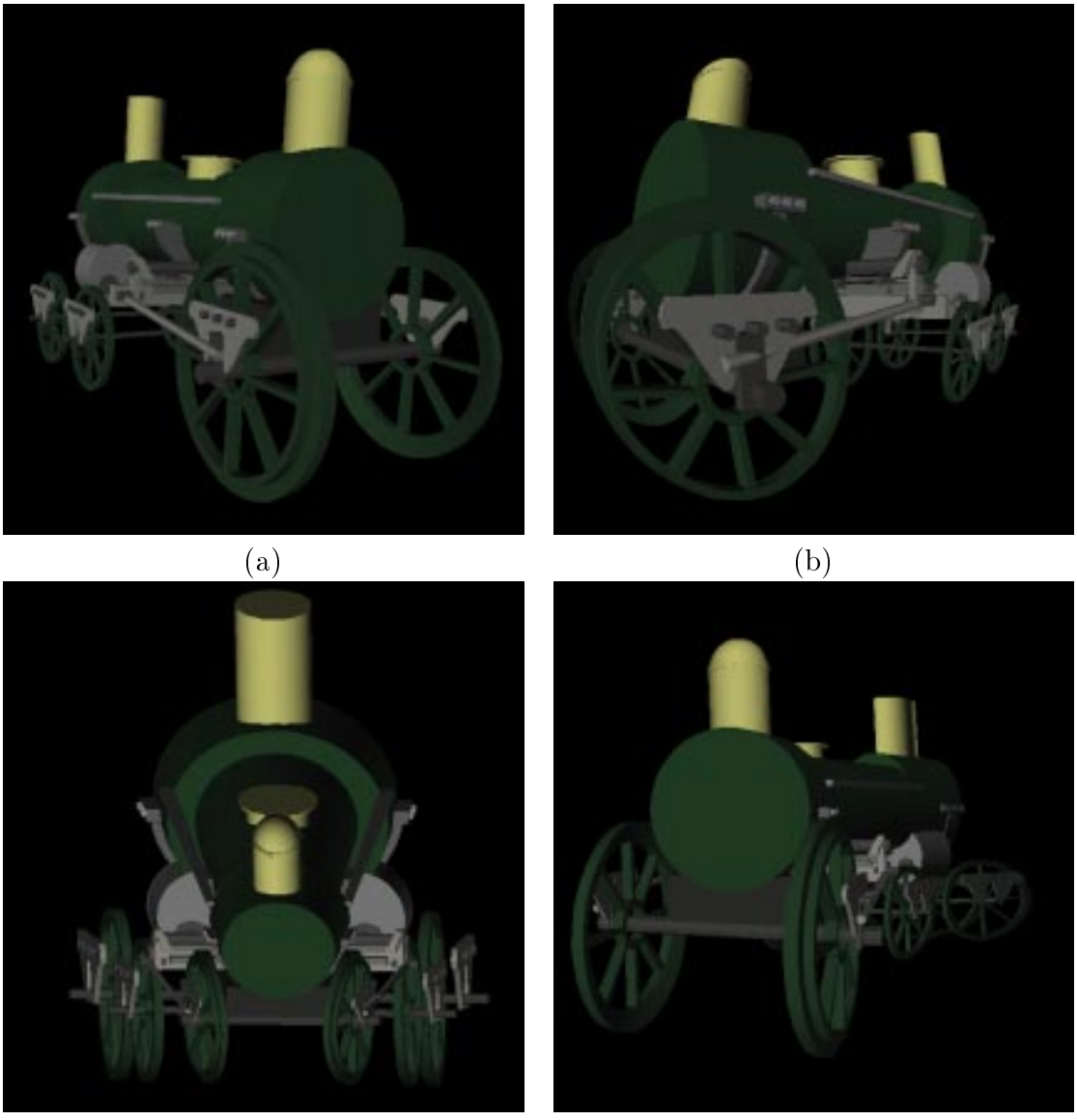
Figure 4-3: The effect of varying the projection surface and C. A test scene has been rendered sixteen times while varying both the curvature of the projection surface and the location of C in the far half of the scene.

Chapter 5

Example renderings

We have applied a variety of expressive projections to three models, as shown in Figures 5-1, 5-2, and 5-3. In Figure 5-1, a model of a nineteenth century locomotive has been projected in four different ways. A planar perspective projection has been included, in Figure 5-1a, for comparison. In Figure 5-1b, where a spherical projection is used, the front left wheel assembly is enlarged while the rear of the locomotive is shrunk in size. This has the advantage of highlighting the wheel, while including the context of the machinery that lies around it. In Figure 5-1c, where an inverse perspective projection is used, the distant parts of the locomotive are spread out. This has the advantage of reducing the degree to which they would be occluded in a conventional perspective rendering, revealing their detail. Finally, in Figure 5-1d, a perspective projection is used where orthogonals curve to the right as they converge. Thus, the front of the locomotive is viewed straight-on, while the rear is viewed from the side. This clearly displays the details of the rear wheels, which would normally be viewed edge-on in a conventional perspective rendering.

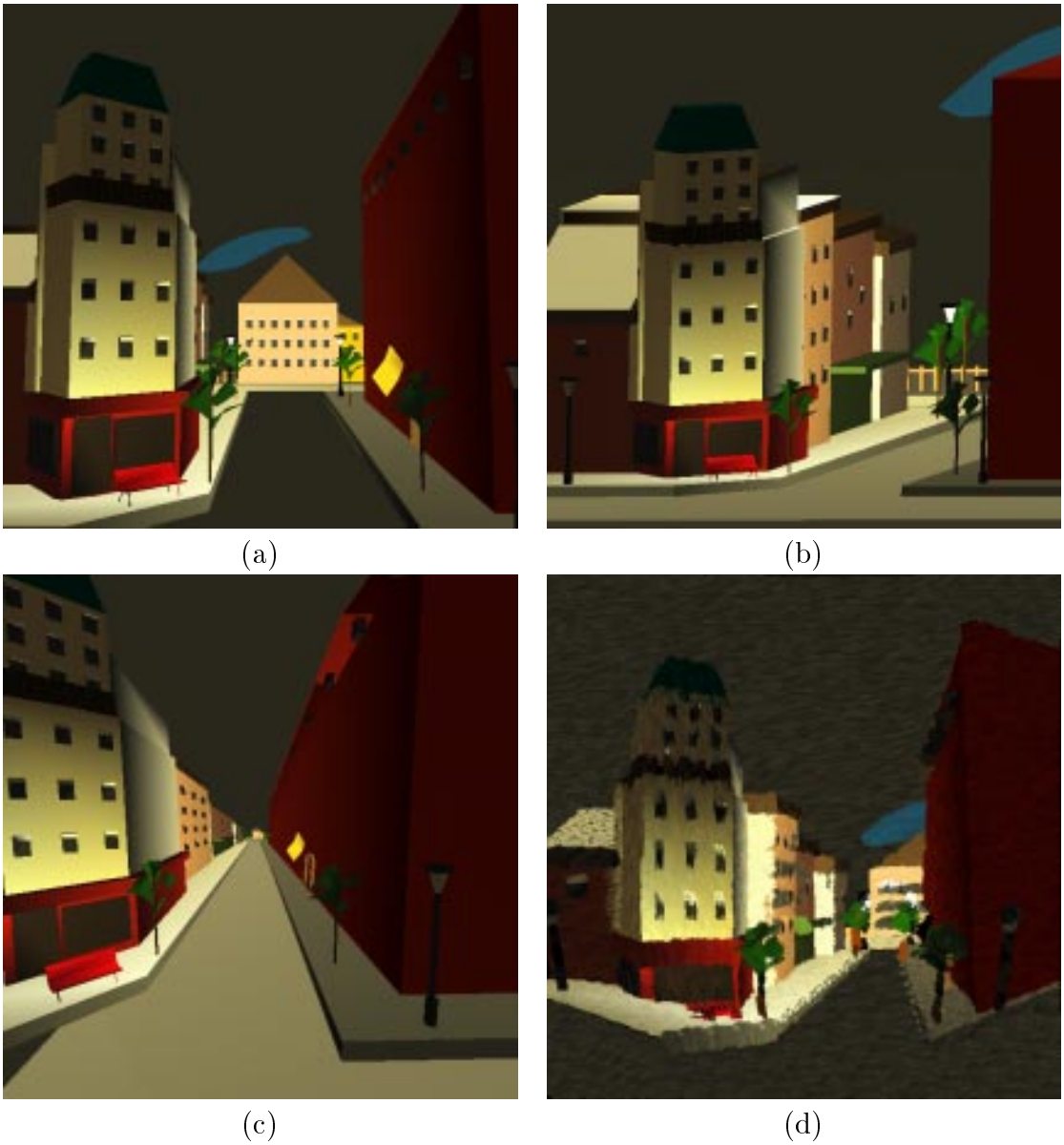
In Figure 5-2, a model of Yves Tanguy's *La rue de la santé* has been projected in four different ways. Once again, a planar perspective projection has been included for comparison. In Figure 5-2b, an oblique projection is used. This has the advantage of highlighting details of the left side of the street. Since distant buildings remain constant in size, detail is not reserved for the foreground or background but is instead spread equally through the depth of the scene. In Figure 5-2c, an accelerated perspec-



(a) (b)

(c) (d)

Figure 5-1: A model of a nineteenth century locomotive projected in four different ways. In (a), the model has been projected in planar perspective; (b) in spherical perspective; (c) in inverse perspective; and (d) in planar perspective, while forcing orthogonals to curve to the right.



(a) (b)

(c) (d)

Figure 5-2: A model of Yves Tanguy's *La rue de la santé (1925)* projected in four different ways. In (a), the model has been projected in planar perspective; (b) in oblique projection; (c) in accelerated perspective (where orthogonals converge twice as quickly); and (d) in spherical perspective, while forcing the orthogonals of the buildings on the left side of the street to curve slightly to the right. This last image was rendered using the author's non-photorealistic painterly renderer fashioned after [14].

tive is used, where orthogonals converge twice as quickly. This extends the range of depth in the scene, making the street and the objects on it appear much longer than they actually are. Aside from giving a sombre aura to the scene, this effect strongly highlights nearby objects and de-emphasizes distant ones. Finally, in Figure 5-2d, a non-photorealistic painterly rendering of the image shown in the transform window of Figure 4-1 is presented. In this rendering, a spherical perspective is used, and orthogonals of buildings on the left side of the street are made to curve slightly to the right. The orthogonals of the building on the right side remain untouched. The spherical projection surface slightly rotates objects in the periphery of the image towards the viewer, while the curved orthogonals are used to pull the distant buildings on the left side out into the street and into view.

In Figure 5-3, a model of a blacksmithing workshop has been projected in four different ways. In Figure 5-3b, the foreground is projected in inverse perspective and the background in perspective (with n varying smoothly between the two). Thus, nearby objects are over-emphasized while distant ones are under-emphasized. This also creates a "swirling" effect, where objects curl smaller as they become more distant, pulling the viewer forward towards the far door and into the back room. In Figure 5-3c, a perspective projection is used, and the vanishing points are pulled to the top left. This reveals the spatial layout of the workshop, clearly displaying the contents of the back room (which would be occluded in conventional perspective). In addition, however, the bellows frame, in green, is projected with a cylindrical surface. This rotates it slightly towards the viewer and reveals more of its side. Finally, in Figure 5-3d, a spherical perspective projection is used. This enlarges the top of the bellows frame, highlighting the crossbar, while maintaining the context of the entire workshop. The bellows frame, however, is projected slightly orthographically (with $n = -0.9$) and is not inertially fitted to the rest of the scene. This over-enlarges it, clarifying its smaller components.

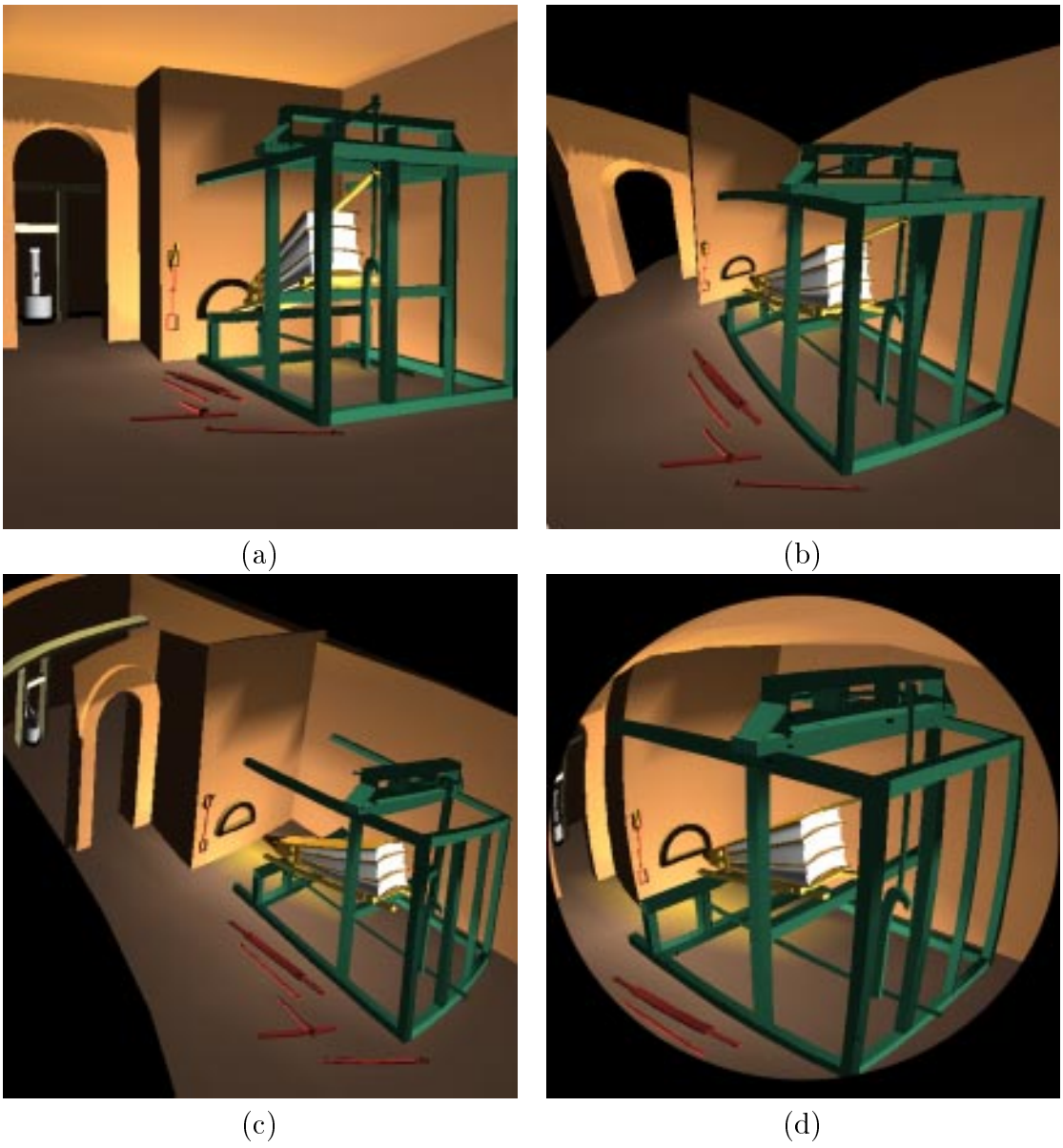


Figure 5-3: A model of a blacksmithing workshop projected in four different ways. In (a), the model has been projected in planar perspective. In (b), the foreground has been projected in inverse perspective and the background in perspective (with n varying smoothly in between). In (c), a perspective projection is used, and the vanishing points are pulled to the top left. In addition, the bellows frame (in green) is projected with a cylindrical surface. Finally, in (d), a spherical perspective projection is used. The bellows frame, however, is projected slightly orthographically (with $n = -0.9$).

Chapter 6

Conclusion

We have presented a framework for interactively computing non-realistic projections for use in NPR systems. The framework provides a means of exploring the use of curved projection surfaces, divergent and/or curved orthogonals, and multiple viewpoints.

Through the course of this work, we have also investigated the visual effects of non-realistic projections and learned how they can address three of the shortcomings of conventional perspective. First, curved projection surfaces are useful for both minimizing wide-angle distortion and for viewing objects in the periphery of an image. Such surfaces can also be useful for highlighting certain objects in the scene by increasing their relative size. Second, controlling the degree to which orthogonals converge in an image is useful for reducing the distortion and ambiguity inherent to the illusion of depth provided by conventional perspective. Finally, providing multiple viewpoints in an image, while controlling whether orthogonals curve as they converge or diverge, is useful for reducing occlusion in an image and revealing the contents of a scene.

Non-photorealistic renderers are proving to be useful tools in both practical and artistic settings. Not only can they be used as a means of controlling how complex information is presented to a viewer, but they can also can be useful as tools for artists to visualize and explore new ideas. Non-realistic projections, in particular, serve as new spatial visualizations. They offer views of 3D space that would be impossible

to obtain via the human eye or any mechanically-realizable devices. When used in combination with traditional media as a means of display, non-realistic projections provide a powerful means of expressing the shape, structure, color, and surface texture of objects. These techniques serve to increase the expressiveness of images, revealing many new and powerful roles.

6.1 Future work

In the short term, two aspects of our system could be improved. First, a more sophisticated solution to the problem of merging multiple projections is required. Instead of trying to position, scale, and orient an image, one alternative might be to incorporate a system of constraints so that certain spatial relationships (such as "on top of" or "adjacent to" relationships) are prioritized and preserved in the image itself. In addition, the new system might also implement a tradeoff between branding an image with the characteristics of its projection and the precision with which certain spatial relationships are constrained.

Second, a more sophisticated representation for the projection surface could be useful. This could potentially increase the range of projections (to include, for example, anamorphic projection). A new representation could also allow for self-occlusion and self-intersection, which have yet to be explored.

In the long term, we see three new avenues for exploration. First, non-realistic lighting has yet to be explored. Altering the way in which light travels through space and interacts with objects could be of significant use. For example, the human visual system uses the magnitude and direction of shadows in an image to deduce spatial relationships and the location of light sources. Highlights in color and intensity are similarly used as visual cues.

Second, non-realistic visibility is also an open topic. Altering which parts of the scene are directly visible and un-occluded could be used to highlight certain objects and force others to be hidden. Twentieth century cubist artists, such as Juan Gris and Georges Braque, have used this technique repeatedly as a means of flattening the

picture surface and breaking the illusion of depth in an image [18].

Finally, the idea of non-realism could be extended into the time domain. We have yet to explore new ways of representing scene changes — including changes in object shape, location, and texture — within either an image or an animation. Recent books on dance notation have begun to explore similar issues, attempting to record movement in a series of images using various visual metaphors [17].

Bibliography

[1] Rudolf Arnheim. *Art and Visual Perception: A Psychology of the Creative Eye.* University of California Press, 1974.

[2] Cassidy J. Curtis, Sean E. Anderson, Joshua E. Seims, et al. Computer-generated watercolor. In *Computer Graphics (SIGGRAPH '97 Proceedings)*, volume 31, pages 421–430, August 1997.

[3] Debra Dooley and Michael F. Cohen. Automatic illustration of 3d geometric models: Lines. *Computer Graphics (1990 Symposium on Interactive 3D Graphics)*, 24:77–82, March 1990.

[4] Debra Dooley and Michael F. Cohen. Automatic illustration of 3d geometric models: Surfaces. In *Proceedings of Visualization '90*, pages 307–314, October 1990.

[5] Julie O'B. Dorsey, François X. Sillion, and Donald P. Greenberg. Design and simulation of opera lighting and projection effects. In *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 41–50, August 1991.

[6] Fred Dubery and John Willats. *Perspective and other drawing systems.* Herbert Press, 1983.

[7] James D. Foley, Andries van Dam, Steven K. Feiner, et al. *Computer Graphics: Principles and Practice, Second Edition.* Addison-Wesley, 1990.

[8] E. H. Gombrich. *Art and Illusion.* Phaidon, 1960.

[9] Paul Haeberli. Paint by numbers: Abstract image representations. In *Computer Graphics (SIGGRAPH '90 Proceedings)*, volume 24, pages 207–214, August 1990.

[10] Masa Inakage. Non-linear perspective projections. In *Modeling in Computer Graphics (Proceedings of the IFIP WG 5.10 Working Conference*, pages 203–215, April 1991.

[11] John Lansdown and Simon Schofield. Expressive rendering: A review of nonphotorealistic techniques. *IEEE Computer Graphics and Applications*, 15(3):29–37, May 1995.

[12] Lee Markosian, Michael A. Kowalski, Samuel J. Trychin, et al. Real-time nonphotorealistic rendering. In *Computer Graphics (SIGGRAPH '97 Proceedings)*, volume 31, pages 415–420, August 1997.

[13] Nelson L. Max. Computer graphics distortion for imax and omnimax projection. In *Proceedings of NICOGRAPH '83*, pages 137–159, December 1983.

[14] Barbara J. Meier. Painterly rendering for animation. In *Computer Graphics (SIGGRAPH '96 Proceedings)*, volume 30, pages 477–484, August 1996.

[15] Takafumi Saito and Tokiichiro Takahashi. Comprehensible rendering of 3-d shapes. In *Computer Graphics (SIGGRAPH '90 Proceedings)*, volume 24, pages 197–206, August 1990.

[16] Thomas Strothotte, Bernhard Preim, Andreas Raab, et al. How to render frames and influence people. In *Proceedings of EUROGRAPHICS '94*, volume 13, pages 455–466, September 1994.

[17] Edward Tufte. *Envisioning Information.* Graphics Press, 1990.

[18] John Willats. *Art and Representation: new principles in the analysis of pictures.* Princeton University Press, 1997.

[19] Georges Winkenbach and David H. Salesin. Computer-generated pen-and-ink illustration. In *Computer Graphics (SIGGRAPH '94 Proceedings)*, volume 28, pages 91–100, July 1994.

[20] Georges Winkenbach and David H. Salesin. Rendering parametric surfaces in pen and ink. In *Computer Graphics (SIGGRAPH '96 Proceedings)*, volume 30, pages 469–476, August 1996.

[21] Daniel N. Wood, Adam Finkelstein, John F. Hughes, et al. Multiperspective panoramas for cel animation. In *Computer Graphics (SIGGRAPH '97 Proceedings)*, volume 31, pages 243–250, August 1997.

[22] Denis Zorin and Alan H. Barr. Correction of geometric perceptual distortion in pictures. In *Computer Graphics (SIGGRAPH '95 Proceedings)*, volume 29, pages 257–264, August 1995.