An Invitation to Discuss Computer Depiction

Frédo Durand

Laboratory for Computer Science, MIT*

Abstract

This paper draws from art history and perception to place computer depiction in the broader context of picture production. It highlights the often underestimated complexity of the interactions between features in the picture and features of the represented scene. Depiction is not always a unidirectional projection from a 3D scene to a 2D picture, but involves much feedback and influence from the picture space to the object space. Depiction can be seen as a pre-existing 3D reality projected onto 2D, but also as a 2D pictorial representation that is superficially compatible with an hypothetic 3D scene. We show that depiction is essentially an optimization problem, producing the best picture given goals and constraints.

We introduce a classification of basic depiction techniques based on four kinds of issue. The *spatial* system deals with the mapping of spatial properties between 3D and 2D (including, but not restricted to, perspective projection). The *primitive* system deals with the dimensionality and mappings between picture primitives and scene primitives. *Attributes* deal with the assignment of visual properties such as colors, texture, or thickness. Finally, *marks* are the physical implementations of the picture (e.g. brush strokes, mosaic cells). A distinction is introduced between interaction and picture-generation methods, and techniques are then organized depending on the dimensionality of the inputs and outputs.

Keywords: Non-photorealistic rendering, computer depiction, perception, visual arts, interaction

1 Introduction

This paper discusses the general problem of *depiction*, that is, the creation of a picture that represents a scene, real or imaginary. It is an attempt to step back and initiate a discussion about the goals and context of computer depiction. There is a variety of picture production purposes, resulting in very different contexts and specificities. We show the complexity and richness of depiction, and the discussion is independent of any implementation. Our main goal is to introduce a vocabulary that will make a principled discussion possible, and to raise questions rather than providing answers. We review and build upon visual arts and perception literature. We outline important issues of depiction that we use to discuss the field of non-photorealistic rendering, and more generally, computer depiction.

Computer graphics has long been defined as a quest to achieve *photorealism*. As it gets closer to this grail, the field realizes that there is more to images than realism alone. Non-photorealistic pictures can be more effective at conveying information, more expressive or more beautiful. The recent field of *Non-Photorealistic Rendering* has developed a wealth of original and effective techniques [GG01, GSS$^+$99, LS95, Rey00, Gre99]. The flip side of this creative explosion is the difficulty of determining the structure of this area and its fundamental goals. These issues were discussed at the recent Symposium on Non-Photorealistic Animation and Rendering [NPA00].

Most authors also agree that the term "non-photorealistic" is not satisfying [NPA00]. The border between photorealism and non-photorealism is fuzzy, and the notion of realism itself is complex [Fer99]. Thomas and Ollie tell an enlightening anecdote about Walt Disney [TJ81], p. 66. Disney would keep asking his animators for more *realism*, which was a cause of misunderstanding, since no one would qualify Disney's animation as realistic. Their interpretation is that he meant *convincing* rather than realistic.

The production of good realistic pictures cannot be reduced to a mechanical recording or, for that matter, to physical simulation. Realistic and non-realistic pictures need to cope with the same issues, and *pictorial* techniques, such as photographic lighting, processing, or dodging and burning, allow the image maker to control expressivity, clarity, and aesthetic, e.g. [Ada95, Apo99].

Moreover, many pictures represent scenes that do not actually exist. The extreme example of impossible figures shows that a picture can superficially look like the representation of a 3D reality, while there is no reasonable objective scene that can be projected to such a picture. This challenges the view where depiction proceeds unidirectionally from an object-space description to a 2D pictorial space.

Artists and other picture makers have developed a rich set

*fredo@graphics.lcs.mit.edu http://gfx.lcs.mit.edu˜fredo

of techniques to produce effective pictures. We believe that computer graphics has much to learn from this large body of knowledge, as well as from the analysis performed in the perception community. The task is not easy because the craft is often elusive or expressed in terms that are not easily translatable to algorithms.

This paper proposes a discussion of computer depiction that encompasses both photorealism and non-photorealism. Non-photorealistic rendering techniques can be different from traditional computer graphics with two respects: They introduce a broader variety of styles and they often offer original computer-human interactions. These differences will be at the heart of the discussion. We discuss the complex interplay between 3D and 2D aspects of depiction, which explains the variety of possible interaction strategies. We also introduce a classification of depiction issues into four systems that provide the basis for a coarse-grain definition of style.

As the title implies, this paper is only a first steps towards a principled discussion of computer depiction. We are working on an extension to this paper, and we hope that articles from other authors will join the discussion. We are looking forward to the reactions and comments of the readers, which will certainly strengthen and broaden the extended version of this article.

1.1 Paper overview

We first discuss vocabulary issues, and place computer depiction in the scope of computer graphics. In section 3, we discuss the complex interplay between the depicted scene and the picture. In particular, we show that depiction involves more than the unidirectional optical projection of a 3D model onto a 2D plane. This explains the variety of both picture styles and interaction strategies. In section 4, we argue that depiction is essentially an optimization problem that aims at producing the most relevant picture for a given purpose. We acknowledge that this optimization problem should most of the time be solved by the user, but the optimization nature of the process requires the design of specific tools for efficient user interaction. In Section 5, we describe a classification of basic depiction issues based on work in perception and art history. Finally, in section 6, we propose a brief review of computer depiction in the light of the previous discussion.

2 Computer depiction

We first discuss the various levels in visual representation. We describe the difference between *image*, *picture*, and *visualization*. We base this discussion on the definition of the Webster dictionary [Web83]. We then place computer depiction in the context of computer graphics.

Image: An image is a "reproduction or imitation", or "the optical counterpart of an object" [Web83]. An image is characterized by optical accuracy to a visual scene or object. The discussion of the various levels of accuracy is beyond the scope of this paper, see e.g. [Hun95, Fer99].

Picture: A picture is "a design or representation," or "a description so vivid or graphic as to suggest a mental image or give an accurate idea of something" [Web83]. The picture is more loosely defined than the image, and it corresponds both to the graphical object and to a representation. In what follows, we use the term "picture" to describe a visual representation of a visual scene, but this representation is not necessarily optically accurate. For example, a line drawing is a picture but not an image. Moreover, as we will discuss, a picture is not necessarily the representation of an existing real scene or object. We can draw pictures of dragons or one-eyed monsters, although none of us has ever seen such animals. Depiction is the production of a picture that represents a scene (real or imaginary).

Visualization: Visualization is "the act or process of interpreting in visual terms or of putting in visual form" [Web83]. The main difference between visualization and depiction is that a visualization can represent visually data or subjects that are not themselves visual. Visualization therefore mainly relies on *metaphors*. Depiction is a special instance of visualization, and realistic image production is a special instance of depiction.

Non-photorealistic: "Non-photorealistic" is a loosely-defined term. It should be used only to qualify a pictorial style. The only meaning of non-photorealistic is that the picture does not attempt to imitate photography and to reach optical accuracy.

Rendering: The field of rendering is concerned with the development of algorithms and numerical methods for the production of pictures given a scene description. Rendering deals with purely automatic techniques and is traditionally not concerned with user interaction.

Non-photorealistic rendering: The field of non-photorealistic rendering has suffered from a loose definition. In particular, it mixes rendering aspects (generation of pictures) together with interaction issues. This is why we advocate the use of a more general term, *computer depiction*.

Computer depiction: Computer depiction deals with all aspects of picture production, and in particular it is concerned with both rendering and interaction. It encompasses both photorealistic and non-photorealistic styles. We will advocate in this paper that most depiction issues are common to realistic and non-photorealistic styles, and that photorealistic rendering is only a special instance of depiction.

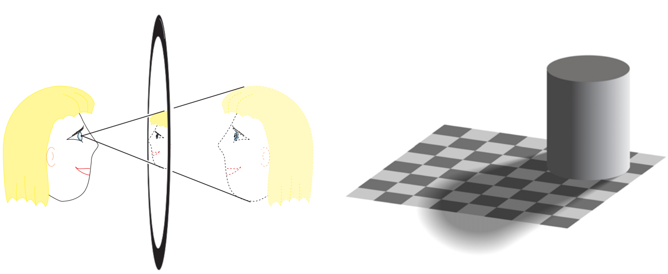
3 From 2D to 3D and back

Traditional computer graphics is a unidirectional projection from a 3D objective scene to a 2D image. The typical object-space inputs are a 3D geometric description of the objects, their material properties and light sources. Perspective matrices, hidden-surface removal, and lighting simulation are then used to project this model onto the 2D image. In this section, we challenge this view, and show that the relation between the object-space scene and the 2D picture can be quite complex, and that picture generation is not unidirectional, but involves many back-and-forth exchanges, feedback, constraints, and goals linking the scene and the picture. This is related to the complexity of the human visual system, and to the dual nature of pictures, both flat objects and representation of an objective scene.

3.1 Intrinsic vs. extrinsic

The notions of *invariant* and *constancy* are crucial in studying vision and the complex dualism of pictures. Invariants are *intrinsic* properties of scenes or objects, such as reflectance, as opposed to accidental *extrinsic* properties such as outgoing light that vary with, e.g., the lighting condition or the viewpoint. Constancy is the ability to discount the accidental conditions and to extract invariants. For example, color constancy consists in discounting the color of the illuminant: We see a red apple as red under illuminant with very different color temperatures, although the physical stimuli have very different objective chromaticities. Size constancy allows us to infer the true size of objects instead of their accidental visual angle: An object does not seem to become smaller when it goes away, because our visual system is somehow able to compensate for foreshortening due to distance.

Constancy is not perfect, but it works surprisingly well. In fact, constancy is usually so efficient that we hardly have conscious access to the extrinsic information present in the retinal image. We do not experience visual angles, we experience objects with their true size and shape. A classical example is when we look at our face in a mirror: We do not realize that the surface of the image on the mirror is half our real size [Gom56] (Fig. 1(a)). Similarly, we can estimate the intrinsic color of an object, but it is very hard to assess the color of the light leaving it (Fig. 1(b)). This is, for example, explained by Land's Retinex theory [Lan77].



(a) (b)

Figure 1: (a) Mirror illusion. The size of our reflection on the surface of a mirror is half our size. (b) In this picture, the white cells in the shadow of the cylinder have the same grey level as the black cells in full light. After an illusion by Ted Adelson.

When looking at a picture, constancy might not operate the same way as when looking at the scene. For example, chromatic adaptation does not function equally. This is why white balance is needed for video cameras, or why different films are required for outdoor photography and for indoor photography without flash. Indeed, when we look at a picture, our visual system adapts to the color of the illuminant of the room in which we look at the picture. In contrast, we are able to discount the intensity of the illuminant in a picture, as demonstrated by Fig. 1(b).

Constancy has caused fundamental difficulties in Western depiction. Constancy is what makes perspective or realistic shading challenging: Because we do not experience visual angles, foreshortening is hard to depict, and because we do not experience absolute extrinsic light intensity but subjective intrinsic lightness, the naive eye is not good at evaluating shading effects. The goal of impressionist painting was to get closer to the transient extrinsic qualities of scenes, and we know how hard an endeavor it was. As noted by Gombrich [Gom56], many realistic painters find it hard to depict a scene without the help of a photograph to visualize the accidental appearance. David Hockney also hypothesizes that painters as early as the 15th century have used optical devices in order to reach realism [Hoc01].

In contrast, other styles produce pictures that are closer to the intrinsic invariants than to the extrinsic appearance. Hogarth [Hog81] tells the anecdote of a Chinese emperor looking at the portrait of a Western king, painted with strong Baroque chiaroscuro (use of light and shade). Commenting on the shadowed half of the face, the emperor asked about the king's disability. For him, a painting represents intrinsic or essential characteristics, and this black half of the portrait had to mean that the king had lost an eye and half of his face. Invariants are often represented directly, not only because invariants are easier for us to consciously access, but also because invariants are by nature a "better," or at least more immutable representation. Some authors strongly believe that the goal of art is the same as the goal of the brain: to extract the *essential* [Zek00, RH99].

The difference can also be stated in terms of depicting "what I see" (extrinsic) as opposed to depicting "what I know" (intrinsic). It suffices to read the opposite statements made by the 19th century painter Turner who claimed, "My business is to paint not what I know, but what I see," and by the 20th century Picasso who declared, "I do not paint what I see, I paint what I know."

In fact, most pictures are hybrid, and managing the balance between extrinsic and intrinsic properties is one of the keys to good depiction. For example, one-point perspective provides an extrinsic view, but preserves the intrinsic orientation of line parallel to the picture plane (horizontals and verticals of the picture). Renaissance chiaroscuro shading renders shapes using light and dark, but emphasizes the intrinsic color, rather than some accidental lighting, as opposed to Baroque tenebrism.

A common way to solve the dilemma between extrinsic and intrinsic characteristic is to choose the depiction such that the extrinsic characteristics match the intrinsic ones. For example, the confusion faced by the Chinese emperor is often avoided by first using a frontal view which preserves the symmetry of the face, and in cinema and photography, by using a *fill light* that illuminates the shadowed areas [Mil91]. Note that this means choosing the depiction situation (constrained viewpoint, additional light source) in order to improve the picture: The 2D picture influences the depicted scene. This is reminiscent of quantum theory and the influence of the observer on the observation. We will come back to these issues.

3.2 Complex mapping

Before discussing further the complex interaction between the picture and the represented scene, and the preservation of intrinsic properties, consider the following striking counter-example to the view of pictures as geometric projections (Fig. 2). When shown a 6-color die, 7-year old children tend to draw it as a single rectangle with 6 vertical or horizontal stripes [Wil97]. The presence of all the colors inside the rectangle rules out the possibility that it may correspond to the projective view of one face. A similar demonstration involves a numbered die: All the numbers are drawn in the rectangle. This demonstrates that the children have mapped the notion of a 3D object with corners, a cube, onto a 2D object with corners, the rectangle.

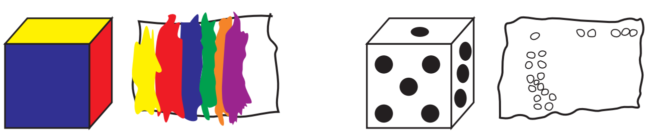


Figure 2: Depiction of a die by children at age 6-7. Redrawn after [Wil97].

This might seem like a very odd example due to the lack of skill. In fact, this is a caricatural but paradigmatic demonstration of a very fundamental principle of depiction: Depiction is not about projecting a scene onto a picture, it is about *mapping properties* in the scene to *properties* in the picture. Projection happens to be a very powerful means to obtain relevant mappings, but it is not the only one, and it is not necessarily the best one.

Consider the drawing of a sphere. Linear perspective projects a sphere onto an ellipse (unless it is in the center of the image). However, most pictures represent off-center spheres as disks, and the projectively correct ellipse is experienced as distorted [Pir70, ZB95]. This is because a perfectly symmetric 3D object should be depicted as a perfectly symmetric 2D object.

We do not advocate abandoning projection matrices. Instead, we suggest that they are only a means, to obtain efficiently a reasonable solution to a much more intricate problem than it seems. And from an epistemological point of view, we should not confuse the means and the end, especially since linear perspective can produce artifacts that cannot be understood from the point of view of projective geometry.

An important issue is the preservation of invariants [Hag86], and whether a given 3D property is preserved by the mapping to the 2D picture. Some systems preserve alignment (e.g. the projective systems commonly used in graphics), some also preserve parallelism (orthographic projection), but for example, perspective does not preserve relative size or the symmetry of spheres.

An interesting aspect of the 2D/3D mapping arises for the line drawing of smooth surfaces. The occluding contour of a surface depends on its differential properties [Koe90]. As illustrated in Fig. 3, convex regions of the surface project as convex outlines, saddle regions project to concave contours, and concave parts can never be represented because they are occluded. However, Willats shows that some artists map the concavity property of the 3D surface to a concave contour in the picture, in order to denote the property of "concavity" [Wil97].

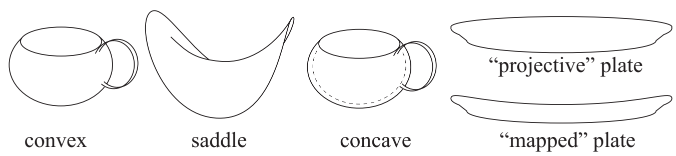


Figure 3: Line primitives and differential geometry mapping. A concave shape, e.g. interior of the cup, is never visible. Nonetheless, some artists choose to depict the concave interior of a plate as a concave 2D contour.

3.3 Primary and secondary space

We now come to the distinction between the expression of representation in the primary vs. secondary space. This point is very counter-intuitive from a computer graphics point of view. The primary space is the 3D objective space, while the secondary space is the picture. This was introduced for the discussion of projection (or drawing) systems [Boo63, DW83, Wil97], and we will extend it to other issues. As we will see, computer graphics has been developed in terms of primary space, while secondary space can provide more flexibility and fits better the mental process of picture production. It can, for instance, explain the difference between one-point, two-point, and three-point perspective, although these three projections fundamentally correspond to the same geometrical operation in the primary space.

The geometry of projection is usually expressed in terms of the intersection between 3D light rays and a picture plane. This is called the *primary geometry* of the projection system [Wil97, Boo63]. It can also be expressed directly in the pictorial space, in terms of *secondary geometry*. Secondary geometry can be seen as a set of rules that teach how to draw various features of the scene, in particular straight lines of the three main axes. For example, perspective projection can be described by stating that distant objects are foreshortened and that orthogonals to the picture plane converge to a vanishing point. Essentially, these are two different descriptions of the same geometrical operation.

Descriptions in secondary geometry are usually less compact than in primary geometry, and harder to adapt to computer graphics. In particular, it is challenging to devise a sufficient and coherent set of generative rules in terms of secondary geometry. However, secondary geometry provides a better account of the mental processes, it permits the expression of a larger variety of drawing and projection systems, and it is more amenable to the description of the evolution in art history and children drawing [Wil97]. Moreover, the complex mapping between 3D and 2D described above are more naturally described in secondary space.

There are two distinct but related difference that make secondary expression more powerful: The expression in picture space makes it easier to express the relation between scene and picture, and the decomposition into a variety of rules for the mapping of various features permits more flexibility. Complex systems often can be described only in terms of secondary geometry. This is the case if only the topology of the scene is preserved, e.g. for subway plans or route maps [AS01]. In this case, drawing is mostly a purely 2D layout problem.

Introducing concepts from secondary geometry is important to provide a larger variety of options, and to design better user interfaces. There is a continuum from pure linear perspective to topological drawing that fit to different purposes, and depending on the context and goals, an expression in primary or secondary geometry will be more useful. And a single technique can mix primary and secondary aspects, such as through-the-lens camera control [GW92], where user interaction in secondary space specifies a camera that is stored internally in primary space.

The distinction between primary and secondary spaces was initially developed to discuss projection systems, but it can be extended to all aspects of depiction. Line drawing is an interesting example. Its primary-space expression is the projection of edges and occluding contours onto the picture plane. However, as shown by e.g. Huffman [Huf71], Clowes [Clo71] and Guzman [Guz71], there is a set of sufficient rules in the picture plane that characterize the line drawing of a 3D objects. These rules in the secondary space describe vertices, edges, T-vertices and end-junctions, and ensure that the direction of occlusion is coherent within the picture. Any picture that respects these rules corresponds to the image of a 3D object. Willats showed that artists intuitively use these rules, and that breaking them results in less realistic pictures [Wil97]. There are a variety of impossible figures based on this: They respect the rules locally, but the global coherence of occlusion is not respected (Fig. 4). It would be interesting to assist the user of a line-drawing system to obtain locally consistent or globally consistent line drawings.

Colors are usually assigned using a primary space specification, through realistic shading and lighting: Incoming light and BRDFs result in the visible color at a given point. In contrast, an example of shading purely in the secondary space occurs in the depiction of a sphere using an illustration software (Fig. 5). A disc is drawn, and a concentric gradient is specified in picture space, resulting in a convincing sphere. Note that the projection is specified in secondary geometry too: The disc is drawn in picture space, regardless of any 3D to 2D projection.

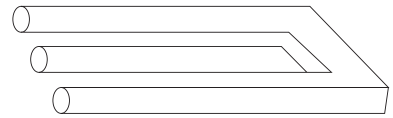


Figure 4: Illusion that respects some secondary space rules of line drawing, but not global occlusion consistency.

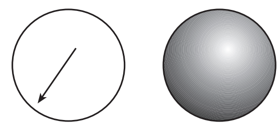


Figure 5: Shading a sphere in picture space.

Similar to the projection, it is fruitful to express realistic shading and lighting in the secondary space and to separate them into various phenomenological rules rather than relying on the more compact rendering equation [Kaj86]. It may seem anti-scientific to break a set of phenomena that can be described by a single compact expression down to a set of phenomenological entities. However, as discussed above, this can provide a better account of the mental process and lead to better user control, and also to a larger variety of styles.

3.4 1kg of 2D, 1kg of 3D, which is heavier?

Our discussion challenges the importance of the primary space. It hints that in many cases, depiction happens mainly in the picture plane. We can go further and wonder about the chicken-and-egg problem between 2D and 3D. Depiction can be seen as a pre-existing 3D reality projected onto the 2D plane, or as a 2D pictorial feature that is superficially compatible with an hypothetic 3D scene. While this may look at first sight like splitting hairs, it reflects very different depiction purposes and contexts. In many cases, the 3D aspects are incidental, and the only significant characteristic lie in the final picture.

Classical computer graphics starts from the 3D model and simulates a view. The typical applications are driving or flying simulation, or architecture rendering, where the fidelity to a given objective scene is paramount. On the other end of the spectrum, illustrations such as the figures in this paper are drawn purely in secondary space, and only imply an hypothetical 3D scene. The depiction of a sphere in Fig. 5 is a good example where depiction is specified only in secondary space. It is a common assumption in graphics that the latter case is an exception, and that most images are projections of 3D scenes.

Consider however the case of movies. For close-ups of dialogues, if the two actors have a different size, the technique of *trenching* is used. A hole is dug in the ground to lower the tallest actor, or the more vertically challenged is put on a box to make their faces level. This means that the 3D scene is altered in order to obtain a good composition in 2D picture space. We are very far from a simulation going unidirectionally from 3D to 2D. A simpler example is group photography: People are asked to take a 3D position that is motivated by visibility issues in the final picture.

Most depiction situations present a mix of 3D and 2D specifications. Acknowledging this richness can result in original techniques that are more relevant to specific contexts. Examples include view-dependent models, where a 3D model is deformed with the only goal of obtaining the desired 2D picture [Rad99, CHZ00, MGT00], or projective drawing that combines the power of 3D notions with the ease of use and flexibility of 2D drawing [TDM01].

4 Depiction as optimization

We have argued that depiction involves complex interactions between the scene and the picture, and that different contexts result in very different depiction strategies. Because pictures always have a purpose, producing a picture is essentially an *optimization process*. Depiction consists in producing the picture that best satisfies the goals. The specification of these goals and the assessment of the quality of the result are obviously intricate issues that go well beyond the scope of computer graphics. Nonetheless, understanding the optimization nature of picture generation has important consequences. This ties up with the previous discussion, in that it invalidates the simple unidirectional projective view of computer graphics.

Vision is an ill-posed inverse problem. It is usually assumed that computer graphics is the corresponding direct image generation, and that it is therefore simple. However, to fully account for the diversity of picture styles and to understand the mental processes involved, one has to think of depiction as the *inverse of the inverse* problem. Indeed, representing a given scene consists in producing a picture that induces a similar impression to beholders as they would have in front of the real scene (Fig. 6). Informally, if we note $V(S)$ the vision operator for a stimulus S, we want $V(S_{picture}) \approx V(S_{scene})$ which means $S_{picture} \approx V^{-1}V(S_{scene})$. If a strict definition is taken for "similar," and if imaging and vision were invertible operations, depiction would be easy and would be reduced to optical simulation.

Unfortunately, vision is a very complicated operator, it is non-invertible since the problem is ill-posed. Moreover, very different stimuli can depict the same scene. For example, a line drawing is a very different optical stimulus from a photograph, but they can as efficiently represent the same scene [RS56].

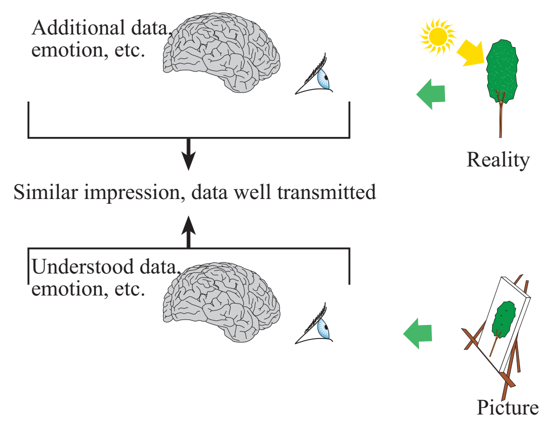


Figure 6: Depiction as the inverse of an inverse problem.

Moreover, pictures have limitations compared to the real optical flow [vH81, BM92]: They are flat, of finite extent, often static, and they have a limited gamut and contrast. These additional constraints are most challenging for realistic images. A very important consequence is that the direct recording of the optical flow (i.e. photography) might not result in the most realistic image. This can be due to, e.g., the absence of depth cues, or to the limited contrast. An image where the contrast at the occluding contour is reinforced might provide a more faithful depth impression, because this compensates for the lack of stereovision or accommodation cues. This is an example of *pictorial techniques* to compensate for the limitation of the medium. A missing cue is rendered through a different perceptual channel (here, stereovision is compensated through occlusion).

Most pictures do not only represent visual properties of the scene. The purpose of the picture can be a message, collaborative work, education, aesthetic, emotions, etc. These additional goals set new constraints on depiction, in terms of clarity, representation of intrinsic vs. extrinsic qualities, 2D layout, etc. Added to the aforementioned limitations of the medium and to the complexity and ambiguities of vision, this results in a very complex optimization problem, where the function to minimize and the degrees of freedom depends heavily on the context and goal. The art and craft of picture creation aims at optimizing the final picture according to a goal, under given constraints set by e.g. the medium, the social context, the artistic fashion. Artists usually do not produce a picture ex nihilo, they work on studies and sketches, and the final picture is retouched until it looks right.

One way to look at realistic graphics is that it is one of the rare cases where the optimization goal (physical accuracy in the primary space) yields a direct analytical formulation of the optimization process. This does not mean that the problem is easy to solve, but at least that it is reasonably simple to state [Kaj86]. We have seen that this might not result in the most realistic picture. Nonetheless, this provides a very close initial estimate, and additional techniques such as model touch-up, photographic lighting, make up, photo processing, can be seen as refinement steps, similar to gradient descent.

In traditional 3D graphics, optimization is dealt with by the user in a feedback loop: The user generates an image, views it, assesses the 2D qualities, reverse-engineers the image, and then performs the hopefully necessary 3D modifications. A new image is rendered, and the process is iterated. We do not propose to replace the feedback loop performed by the user by optimization software. In most situation, this would prove impossible because of the difficulty to translate and solve for artistic goals. Moreover, many users want to keep control of the process. However, there are cases where software optimization proves useful, e.g. [AS01, GRMS01, SL01, Hau01, KS00, GFMS95, Her01].

Our argument is at a more "philosophical" level: We need to recognize the complexity of the depiction problem and its optimization dimension in order to develop relevant solutions. There are essentially three strategies to solve this optimization problem: The user can solve it, the computer can solve it, or the solution might involve both user and computer decisions. All approaches are of course not contradictory and can be blended. Each strategy raises a number of issues, which we only briefly outline.

If the user solves the optimization and basically explores the parameter space:

- Provide relevant degrees of freedom in the rendering algorithm, e.g. [Bar97].
- Linearize parameter space. In particular, the controls should be predictable and uniform, that is, a small change in a parameter should result in a predictable change, and the perceptual magnitude of the change should be uniform. Good examples are the CIE-LAB color space [Fai98] or the perceptually-uniform gloss model by Pellacini et al. [PFG00].
- Provide controls in image space to control the primary space (inverse kinematics [Par01], painting with light [SDS$^+$93], through-the lens camera control [GW92]).
- Provide high-level controls directly related to the goals and constraints of the user, e.g. [DOM$^+$01].
- Develop purely secondary-space pictorial techniques. Since the standard "projection" is often close to the desired solution, a small perturbation is often enough to obtain the desired picture. Examples include digital dodging and burning tools or tone mapping, e.g. [Tum99].
- Speed up the internal loop to provide faster feedback to the user, e.g. [GKR95, GH00].

Design galleries is a typical tool to help users explore a complex parameter space. The computer performs all the computations based on the primary parameter space, and presents a choice to the user based on the secondary characteristics of the output [MAB$^+$97].

If the computer solves the optimization:

- Define the energy function. This involves cognitive psychology and understanding of traditional techniques, e.g. [SF91, HCS96, AS01, SL01].
- The traditional optimization issue: exploration of a highly non-linear parameter space. This ties up with the need for predictable and uniform parameter spaces.

The general case is mixed. The computer has to take decisions automatically, but the user wants to keep some control and influence the decisions. This is for example one of the exciting issues raised by the convergence of games and movies: The computer has to respond automatically to the user interaction, but the equivalent of the movie director want to keep control of the style of pictures. The technique by Hertzmann is an example where the user has some high-level control on stylistic parameters [Her98]. The amount of user vs. computer control is an exciting issue in designing computer depiction systems.

5 Organizing computer depiction

The difficulty in classifying and comparing non-photorealistic rendering techniques is parallel to the difficulty faced in picture studies to discuss very different styles of pictures. This is why we introduce and adapt the classification developed by Willats [Wil90, Wil97]. He builds upon various fields to propose a structural study of representation that encompasses not only fine art from all eras and civilizations, but also any kind of picture, be it a child's drawing, a traffic sign, a repair manual or a logo. His initial goal was to provide a structured language to describe "how" these pictures are different. While introducing new vocabulary, he notes that "Physics as a science simply did not exist before the introduction of a precise terminology (...) New words had to be introduced for new concepts and old words (...) had to be given a new and precise meaning" [Wil97], page 5.

5.1 Representation systems

The central thesis of Willats is that depiction (or representation in his terms) can be described in terms of two systems: the *drawing* systems and the *denotation system*. In Willats's words, "the drawing systems are systems such as perspective, oblique projection and orthogonal projection that map spatial relations in the scene into corresponding relations in the picture" [Wil97], page 2. "The denotation systems map

(...) *scene primitives* (...), into corresponding *picture primitives*, such as regions, lines, or points," [Wil97], page 4. To summarize Willats's theory, depiction involves two kinds of decisions: which primitives to use (denotation), and where to put them (drawing).

The term "drawing" in Willats's classification introduces potential ambiguities, because it is used to describe both the spatial system and the line-drawing denotation system (use of 1D primitives). This can be explained by the historical role of line drawing in Western art. It was used for studies of paintings, in order to find the right composition and the right spatial mapping for the various features. It is only recently that line drawing has acquired the status of art form. We use the term "spatial" instead of "drawing," and "primitive" instead of "denotation," because these terms carry less ambiguity.

We extend Willats's framework, and we decompose depiction into four kinds of systems: spatial, primitives, attribute, and marks. An information processing point of view would state that direct picture production goes through a pipeline of 4 stages: spatial mapping, choice of primitives, attributes of these primitives, and mark implementation. However, we have shown that picture production is not always direct and that the mapping involved can be intricate. The pipeline metaphor is only meaningful in the very particular case of the mechanical rendering from 3D to 2D.

Spatial system: The spatial system deals with the spatial properties of the picture. In the case of direct image generation, it maps 3D spatial properties to 2D spatial properties. Note that the mapping can be implicit, in particular when the picture does not represent a real 3D scene.

In traditional computer graphics, the spatial system is handled by projective matrices that project 3D coordinates onto 2D picture coordinates. However, more elaborate spatial systems have been used, e.g. non-linear perspective [BFR95, LG96], multiple perspectives in a single image [AZM00], or purely topological spatial layout [AS01].

Primitive system: The primitive system maps primitives in the object space (points, lines, surfaces, volumes) to primitives in the picture space (points, lines, regions). In contrast to Willats's classification, we introduce the distinction between continuous and discrete point primitives. A discrete point primitive is for example the symbol representing a station in a subway map, while the pixels in a ray-traced image are continuous point primitives.

The primitive system has long been neglected because the traditional systems are trivial. For example, in classical computer graphics, the primitive system maps visible points in the scene to point primitives in the image. Willats calls this *optical* denotation. In the *line drawing* primitive system, 1D lines in the picture denote silhouettes of the scene. Silhouette extraction is the main primitive issue in NPR, e.g. [EC90, Goo98, MKT$^+$97, ZH00, ST90, RC99, Cur98, BS00]. There are also non-trivial primitive systems, for example ball-and-stick drawing, where an elongated volumetric cylindrical shape such as an arm is mapped to a line primitive (Fig. 7) [Wil97, HOT98].

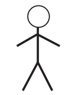


Figure 7: Ball-and-stick drawing of a man.

Attribute system: The attribute system assigns visual properties such as color, texture, thickness, transparency, wiggleness, or orientation to picture primitives. The list of relevant visual attributes depends on the primitive, on the mark system and on the context (see below).

Willats discusses attributes only for the optical denotation system (continuous point primitive), but attribute issues occur for all primitive systems. It is, for example, common in line drawing to assign the color and thickness of strokes to depict shading. In realistic graphics, the attribute system is physically-based lighting and shading. Recent work on attribute systems include [Wil91, CJTF98, GGSC98, SMGG01, GSG$^+$99].

Mark system: The mark system is the implementation of the *primitives* placed at their *spatial* location with the corresponding *attributes*. The mark system describes the physical strokes in traditional depiction, and in rendering, it is responsible for medium simulation (e.g. oil painting, pencil brush, watercolor, engraving).

Traditional computer graphics simply uses pixels as marks, and the correspondence between primitives and marks is direct. Nevertheless, there is a fundamental difference between picture primitives, and marks that are only the physical implementation of primitives. For example, a line primitive can be implemented as a series of dot marks, e.g. in a mosaic, and a paintbrush can be used to implement either 1D long brush strokes or 0D pointillism. Many mark techniques have been presented in NPR, e.g. [CAS$^+$97, SB99, Ost99].

5.2 Depiction cannot be reduced to systems

The systems presented above permit a principled coarse-grain decoupling of depiction issues. They are crucial to understand the various aspects of depiction. Nevertheless, it is equally important to discuss the complex interaction between these systems, and the inherent limitations of the decomposition of depiction into sub-tasks. This framework does not provide a strict and complete classification, due to the richness and complexity of the endeavor.

We first insist that these systems can be more complex than a simple projection from 3D to 2D. They assign mappings between the object space and the picture space. The mappings can be non-trivial (as in the die example Fig. 2 or as for the plate in Fig. 3). They can also be implicit, from 2D to 3D when depiction is specified purely in picture space as in Fig. 5.

There can be very rich interactions between the various systems. The simplest interactions between systems is the constraints set by one system on another. In particular, the mark system imposes constraints on the range of possible primitives and attributes (color gamut, physically-possible thickness). Each primitive also comes with a different set of attributes. Thickness, for example, is not relevant for regions or continuous points.

But there can be less mechanical interactions as well. For example, the decision to include a primitive (such as a line in line drawing) might depend on its spatial proximity with other primitives to avoid cluttering. And the balance and composition of an image relies on the spatial layout, on the arrangement of colors and intensities, but also on the saliency of various primitives. The art and craft of effective picture making relies on the rich and complex interaction between all aspects.

Consider for example the variation of color inside a single mark such as oil painting or watercolor. Depending on the point of view, this can be viewed at three different levels. It can be directly specified by the attribute system. It can also be a simple stroke texture purely at the mark level. It can also be partially controlled by the attribute system via a variance of color attribute that controls the amount of color variation inside a stroke Hatching is another example where the continuous point and line drawing primitive systems interact in a very intimate and rich way with the stroke mark system. The same mark primitive is used to implement both hatching and silhouettes [SABS94, WS94, SWHS97, DOM$^+$01], and in master's drawing, it is hard to tell one from the other.

Decomposing a given picture into these four sub-systems can be ill-posed. However, they provide a vocabulary to discuss basic techniques and to relate computer depiction to traditional picture production.

5.3 Classification

Now that we have introduced important issues and vocabulary, we are about to present a brief survey of low-level computer depiction techniques, focusing on technique categories rather than on depiction style. This survey is partial because the domain is vast, but we hope that it outlines major issues. NPR research is usually organized according to the kind of systems (interactive, automatic, 2D or 3D) or depending on the simulated media. These classifications are useful and correspond to some of the issues discussed above. However, we believe that it is also important to decompose computer depiction software into lower-level modules performing precise tasks. This is crucial to permit the cross-integration of different techniques, and to provide a better account of the potential of each method.

We use our classification of representational issues: spatial, primitive, attributes, and marks. Techniques can then be classified according to their representation *style* and to their inputs/outputs. In this paper, we focus on the inputs and outputs. For each technique, the main inputs can be in 3D primary space, or in 2D secondary picture space, or hybrid: e.g. z- or G-Buffer, which we will denote loosely as $2.5D$. This classification is related to the difference between object precision and image precision [SSS74], and to the difference between discrete and continuous representations.

We will use a simple notation $nD \rightarrow mD$ to describe a technique with inputs in the n-dimensional space and output in the m-dimensional space. A method can have two distinct goals: actual picture generation or interaction. Note that by interaction, we not only mean user interaction, but also computer-aided techniques such as optimization that take depiction decisions. Straight picture generation globally goes from 3D to 2D (but can also use some $3D \rightarrow 3D$ or $2D \rightarrow 2D$ techniques). In contrast, interaction can include some direct 3D or 2D manipulation, but may also include feedback from $2D$ to $3D$. For example, the through-the-lens camera control allows a user to control the 3D camera using 2D interaction [GW92].

This classification according to the dimensionality of the inputs and output accounts for the recent diversity in interaction strategies, and permits the discussion of recent subfields such as image-based modeling ($2D \rightarrow 3D$) and rendering ($2.5D \rightarrow 2D$), or sketch-based modeling ($2D \rightarrow 3D$) .

Additional important criteria are whether the method sets absolute or relative properties, and if it is global or local. For example, lighting and shading set absolutes color values, while atmospheric perspective is more a relative modification of the color. And methods can be used globally on the whole picture, or vary spatially, or be limited to a subset of objects in the scene.

Some complex techniques might prove hard to fit strictly in our classification, in that they involve intimate coupling between different systems. As we have seen, depiction is quite an intricate endeavor, and it is unlikely that a single framework will rigidly account for the variety of solutions. However, our classification provides a vocabulary and a reference to discuss such complex or original systems. We believe that a principled discussion of basic techniques is a necessary step to be able to discuss more complex solutions, and we encourage readers to devise new depiction styles and new interaction solutions, by building upon our classification or by building upon the limitations of our classification. Moreover, computer depiction should not be limited to the imitation of traditional techniques and media, but has the potential to produce novel forms of depiction.

6 A tentative overview

In what follows, we simply illustrate the descriptive potential of our framework and discuss examples of work in the various categories. The discussions are unfortunately brief, and are not meant as a comprehensive survey. Instead, they are provided as additional illustrations of the concepts discussed so far. This section is less conversational, and is more intended as the skeleton of a larger discussion. We invite the reader to pursue the reflection along those lines, and we are working on an extended version of this paper.

6.1 Spatial

Traditional $3D \rightarrow 2D$ spatial techniques include linear perspective and orthographic projection expressed in primary geometry, e.g. [CP78]. Non-linear spatial systems have also been used [Max83, Grö94, LG96, GG99, BFR95, Lev98, AZM00, RB98, Gla00, CT01].

On the other hand, $2D \rightarrow 2D$ techniques consist in warping within the picture plane [GDCV98, Lit91]. A good illustration of $2D \rightarrow 2D$ spatial system is the reprojection of panoramas (curvilinear perspective) to obtain linear perspective views, e.g. [Che95, TDM01]. The method by Zorin and Barr corrects for perspective distortion using a $2D \rightarrow 2D$ technique to preserve either alignment or sphere symmetry [ZB95]. Seitz and Dyer present another $2D \rightarrow 2D$ spatial technique that is particularly interesting because it occurs purely in the secondary space, but respects an hypothetical 3D geometry [SD96].

Perspective has been a subject of intense debates in the visual arts [Hag86, Pir70, Pan27, Kub86, Kem90, Elk94]. The difference between $3D \rightarrow 2D$ issues – basically visibility – and $2D \rightarrow 2D$ issues is often overlooked and results in misunderstandings between parties. Some authors argue that linear perspective is “natural” and respects human vision because it faithfully respects visibility, that is, it is the projection from a given point [Gom82]. However, they miss the fact that any 2D warping of a linear-perspective image also respects visibility. Expressed in primary geometry, it means that the projection with respect to a point can be performed on any manifold, e.g. plane, sphere or cylinder. We have seen that it is more fruitful to state the debate in terms of invariant or property preservation. In this case, there is no perfect “natural” solution since we cannot preserve both linearity and the symmetry of spheres.

Interaction techniques going from 2D inputs to 3D can be used to control the camera [GW92, EHW97]. Injecting more secondary geometry controls in Agrawala et al.’s multiperspective technique [AZM00], as well as in other non-linear perspective work [LG96, GG99, BFR95, Lev98, RB98, Gla00] would greatly improve their usability.

Another class of $2D \rightarrow 3D$ interactions facilitates the modeling phase. Approaches have been proposed to sketch 3D objects using 2D strokes [ZHH96, IMT99, CHZ00], or to build 3D models from photographs, e.g. [FLR$^+$95, DTM96]. Similar $2D \rightarrow 2.5D$ techniques also exist [HAA97, OCDD01, CT01, ZDPSS01]. Other hybrid interactive spatial systems allow a user to draw in 2D but modify the view or move objects in pseudo-3D [TDM01, BCD01]. And as discussed above, view-dependent models allow 2D spatial objectives to control 3D models [Rad99, MGT00]. Gooch et al. [GRMS01] use optimization to choose the 3D camera parameters with a 2D goal: good composition.

Optimization has also been used to solve 2D spatial aspects. Agrawala et al. compute route maps using 2D optimization loosely respecting the 3D geometry according to cognitive findings [AS01]. It is particularly interesting to note that their approach is based on shape properties (length, angle) and not directly on spatial coordinates. Graph drawing is also a pure 2D optimization problem [BETT99], and recently, Escherization optimizes a shape to tile the plane [KS00]. An extension to optimizing over the 3D domain would be quite exciting.

Finally, we discuss the multiperspective cell panorama technique by Wood et al. [WFH$^+$97]. This technique is very interesting because from the point of view of the beholder, it looks like linear perspective, while for the artist, it requires a highly non-linear spatial system. The authors noted that Disney’s artists were able to produce more convincing multiperspective panorama than the computer-assisted method. We hypothesize that this is because their spatial system operates on the primary geometry. In contrast artists reason only in terms of secondary geometry, which alleviates them from the constraints of primary geometry, and allows them to think directly in terms of goals and property mapping. An exciting subject of future work would start from the automatic primary-geometry solution, and use relaxation to optimize the multiperspective, in order to minimize distortions in terms of secondary geometry.

6.2 Primitive

Recall that there are four different kinds of picture primitives: continuous points, discrete points, lines and regions. 1D primitives probably yield the richest variety of denotation systems. Lines can denote a large class of scene primitives. They can be classified into view-independent and view-dependent primitives. View independent primitives include very thin objects (such as strings), elongated objects (such as legs), edges of objects, reflectance discontinuity (such as the limit of a patch on a cow), shadow boundaries, or transparency edges.

View-dependent 1D primitives consist in occluding contour, and a special case of occluding contour, the external silhouette of objects, and the limits of specular highlights. The latter is a case where the denotation system (line drawing of the highlight) interacts with the attribute system (shininess of the material). Shadows raise similar issues, since they can depend on the primitive, attribute and spatial systems.

We distinguish three approaches to silhouette extraction $3D \rightarrow 2D$ [EC90, Goo98, MKT$^+$97, SGG$^+$00, ZH00], $2.5D \rightarrow 2D$ [ST90, RC99, Cur98, BS00] or $2D \rightarrow 2D$ [Can86, PHM90].

An important issue of future work is the design of edge selection algorithms. Artists have the ability to draw only the relevant edges to depict an object. This can be addressed by devising selection rules, or interactive selection tools.

Elder et al. propose to adapt image editing to work in what they call the *contour domain* [EG01]. An image is represented as a set of edges and continuous smooth regions. Their technique basically transforms the image from a continuous-point representation, to a 1D-line representation, which facilitates some editing operations for the user.

6.3 Attribute

The set of possible picture attributes depend on the primitives, marks, and on the context. Attributes include color, tone, transparency, texture, thickness, wiggling (for lines primitives, e.g. [FS94]), or orientation. Color can be expressed in different color spaces, such as RGB, HSV, or other dimensions such as cool-to-warm can be used, e.g. [GGSC98].

The classical $3D \rightarrow 2D$ attribute system is lighting and shading, where illuminance and BRDF are combined to compute a visible color. The non-photorealistic technique by Gooch et al. is particularly interesting because it uses the intrinsic color of the objects with a relative extrinsic lighting mapped on the cool-to-warm dimension [GGSC98]. Shading methods have been introduced for line primitives as well, e.g. [SST89, TTT91, GSG$^+$99]. Atmospheric perspective is a very interesting pictorial technique where the distance can be mapped to different attributes. It can affect saturation, make distant objects more bluish, decrease sharpness, etc. In fact, the most important aspect of aerial perspective is to *group* parts of the scene at a similar distance by assigning them a common property.

$2D \rightarrow 2D$ attribute techniques include standard image controls such as contrast/brightness or color modification, e.g. [RAGS01], or dodging and burning [Ada95]. Tone mapping is also a $2D \rightarrow 2D$ attribute technique, that specifically copes with the limitations of the medium, e.g. [TT99]. Note that the same pictorial effect – decreasing the contrast – can also be obtained in a $3D \rightarrow 3D$ way, using appropriate lighting [Mil91].

Hybrid approaches include works such as shading in two dimensions [Wil91], the commercial product ZBrush [ZBr], and the comprehensive rendering of shape [ST90].

The case of graftals is a rather complex attribute system, since it heavily interacts with denotation and marks [KMN$^+$99, MMK$^+$00]. Complex materials such as fur or plants are rendered using procedurally-generated strokes. Their work moreover permits high-level as well as spatially-varying graftal style specification, and can be used in conjunction with lighting and shading.

As mentioned before, attributes can be modified by altering the 3D scene. Examples of $3D \rightarrow 3D$ attribute techniques include photography lighting [Mil91] or make up [Auc99].

$2D \rightarrow 3D$ interaction techniques permit the deduction of lighting from desired color [SDS$^+$93], or from sketching highlights and shadow [PF92, PRJ97]. In the recent lit-sphere method, an artist paints an example sphere, which is remapped to an environment map for 3D rendering [SMGG01]. 3D painting [HH90, ABL95, 3D] allows a user to edit the texture maps of the object-space model using a 2D interface. Image-based editing systems offer similar possibilities [SK98, OCDD01] and can be classified as $2D \rightarrow 2.5D$.

$2D \rightarrow 3D$ interaction techniques have been developed to inject some 3D attribute notions into an otherwise purely two-dimensional depiction context. This includes texture mapping for cell animation[CJTF98], and shadows for cell animation [PFFL00] or for architectural sketching [TDM01].

We finish this overview of attribute systems with a discussion of orientation, e.g. [Hae90, SWHS97, ZH00, Hau01]. While orientation is used to drive the mark system, it is important to consider it as an attribute, since orientation is a general issue, common to a variety of mark styles. Separating the orientation issue from the particular marks is crucial to build generic modules. Most mark system using orientation attributes can be used to display any 2D vector field, e.g. pen and ink strokes [SWHS97], streamlines [TB96], or LIC [CL93]. $3D \rightarrow 3D$ orientation computation have been proposed using iso-parametric curves [Elb95] or principal curvatures [ZH00]. $3D \rightarrow 2D$ [ST90]. Similarly, $2.5D \rightarrow 2D$ exist, e.g. [RK00]. The $2D \rightarrow 2D$ category offers both automatic [Hae90, Hau01] and user-controlled [Hae90, Ost99, SWHS97] approaches.

6.4 Mark

The mark system is the last representational aspect. It deals with the physical medium of the picture. The mark system can be trivial, in particular for realistic graphics where the mark is simply a pixel. On the other end of the spectrum, photomosaics use pictures as marks, and some very advanced physically based simulation have been developed for various media, e.g. [CAS$^+$97, SB99, TFN99, BSLM01, GM97].

The mark system is a special case, because it does not really involve a 3D to 2D mapping, which has been dealt with by the previous systems. The mark system is thus mainly a 2D problem. However, we will see that in some cases, especially for animation, 3D aspects are important.

The example of halftoning [Uli87] is paradigmatic of mark system because the input, output and specification are clearly defined. Halftoning takes as input a grey-scale (or color) image and translates it into a binary image that provides the viewer with a faithful tonal impression. Central to halftoning is the linearity of the reproduction curve, which more generally means that the output should be predictable from the input. Halftoning has also been extended to richer patterns [OH95, VB99].

Optimization has been used for mark systems, either to optimize their location [Hae90, Hau01, DHvOS00], or tonal fidelity [OH95, Ost99].

One of the challenges for mark systems is raised by NPR animation, where mark coherence is paramount. Approaches based on the 3D geometry [Mei96, Cur98, PHMF01] or motion flow have been proposed [Lit97, HP00]. So far, the most successful approaches have used a combination of 3D-based coherence with picture-based criteria [Mei96, Cur98, PHMF01]. Again, the interplay between 3D and 2D is at the heart of the richness and complexity of depiction.

7 Invitations

We hope that this article will stimulate discussion and future work in computer depiction. We insist that the framework proposed in this paper is not intended as a rigid set of boxes for the sake of classification. We hope that it provides a vocabulary and raises issues. We are also well aware that, due to the complexity of depiction, different classification can be proposed along dimensions similar or orthogonal to the ones discussed in this paper.

The extension to animation is far from straightforward. We have seen that 2D pictures are not a simple section of the optical flow. Similarly, they are not a simple cross section of space-time.

We want to study existing NPR software in this framework, and describe their *image generation* and *interaction* work flows. This should highlight similarities, potential cross-integrability, as well as original designs that can be applied to different problems. The design of a versatile NPR system implementing the diversity of depiction styles and interactions is a challenging task, raising both software design and depiction issues.

Higher-level issues need further discussion. This include notions of abstraction, precision, selection, as well as aesthetic issues such as composition, balance or color harmony.

The classification into four depiction systems provides a structure for a coarse-grain definition of style. The refinement of this definition raises exciting issues in stylistics, and could allow us to parameterize, capture and reuse style.

The availability of this new variety of styles raises the important question of the choice of an appropriate style, especially when clarity is paramount. These are cognitive psychology questions, but we hope that computer depiction can provide both an experimental testbed, and theoretical hints.

Acknowledgment

The careful comments of the reviewers greatly helped to improve the paper. Many thanks to Julie Dorsey, Victor Ostromoukhov, Pat Hanrahan, Maneesh Agrawala, Fabrice Neyret, Joëlle Thollot, Byong Mok Oh, and the students of the 4.209 course *The Art and Science of Depiction*. This research was partially funded by an NSF CISE Research Infrastructure Award (EIA-9802220).

References

[3D] Deep Paint 3D. http://www.deeppaint3d.com.

[ABL95] M. Agrawala, A. Beers, and M. Levoy. 3d painting on scanned surfaces. *ACM Symp. on Interactive 3D Graphics*, 1995.

[Ada95] A. Adams. *The Camera+The Negative+The Print*. Little Brown and Co, 1995.

[Apo99] A. Apodaca. Photosurrealism. In A. Apodaca and L. Gritz, editors, *Advanced Renderman : Creating CGI for Motion Pictures*. Morgan Kaufmann, 1999.

[AS01] M. Agrawala and C. Stolte. Rendering effective route maps: Improving usability through generalization. In *Computer Graphics (Proc. SIGGRAPH)*, 2001.

[Auc99] K. Aucoin. *Making Faces*. Little Brown and Co, 1999.

[AZM00] M. Agrawala, D. Zorin, and T. Munzner. Artistic multiprojection rendering. In *Eurographics Workshop on Rendering*, 2000.

[Bar97] R. Barzel. Lighting controls for computer cinematography. *J. Graphics Tools*, 2(1), 1997.

[BCD01] D. Bourguignon, M.P. Cani, and G. Drettakis. Drawing for illustration and annotation in 3d. *Computer Graphics Forum*, 20(3), 2001.

[BETT99] G. Di Battista, P. Eades, R. Tamassia, and I. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, 1999.

[BFR95] V. Bourgoin, N. Farenc, and M. Roelens. Creating special effects by ray-tracing with non classical perspectives. Technical Report 1995-13, École Nationale des Mines de St Etienne, 1995.

[BM92] Barbour and Meyer. Visual cues and pictorial limitations for computer generated photorealistic images. *the Visual Computer*, 9, 1992.

[Boo63] P. J. Booker. *A History of Engineering Drawing*. Chatto and Windus, 1963.

[BS00] J. Buchanan and M. Sousa. The edge buffer: A data structure for easy silhouette rendering. In *Non-Photorealistic Animation and Rendering*, 2000.

[BSLM01] B. Baxter, V. Scheib, M. Lin, and D. Manocha. Dab: Interactive haptic painting with 3d virtual brushes. *Proc. SIGGRAPH*, 2001.

[Can86] J. Canny. A computational approach to edge detection. *IEEE Trans PAMI*, 8(6):679–698, 1986.

[CAS+97] C. Curtis, S. Anderson, J. Seims, K. Fleischer, and D. Salesin. Computer-generated watercolor. *Proc. SIGGRAPH*, 1997.

[Che95] E. Chen. Quicktime VR - an image-based approach to virtual environment navigation. *Proc. of SIGGRAPH*, 1995.

[CHZ00] J. Cohen, J. Hughes, and R. Zeleznik. Harold: A world made of drawings. In *Non-Photorealistic Animation and Rendering*, 2000.

[CJTF98] W. Toledo Corrêa, R. Jensen, C. Thayer, and A. Finkelstein. Texture mapping for cel animation. *Proc. SIGGRAPH*, 1998.

[CL93] B. Cabral and L. (Casey) Leedom. Imaging vector fields using line integral convolution. In *Proc. SIGGRAPH*, 1993.

[Clo71] M. Clowes. On seeing things. *Artificial Intelligence*, 2(1):79–116, 1971.

[CP78] I. Carlbom and J. Paciorek. Planar geometric projections and viewing transformations. *Comp. Surveys*, 10:465—502, 1978.

[CT01] S.-H. Chu and C.-L. Tai. Animating chinese landscape paintings and panorama using multi-perspective modeling. In *Computer Graphics International*, 2001.

[Cur98] C. Curtis. Loose and Sketchy Animation. In *SIGGRAPH: Conference Abstracts and Applications*, 1998.

[DHvOS00] O. Deussen, S. Hiller, C. van Overveld, and T. Strothotte. Floating points: A method for computing stipple drawings. *Computer Graphics Forum*, 19(3), 2000.

[DOM+01] F. Durand, V. Ostromoukhov, M. Miller, F. Duranleau, and J. Dorsey. Decoupling strokes and high-level attributes for interactive traditional drawing. In *Eurographics Workshop on Rendering*, 2001.

[DTM96] P. Debevec, C. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *Proc. of SIGGRAPH 96*, 1996.

[DW83] F. Dubery and J. Willats. *Perspective and Other Drawing Systems*. Van Nostrand, 1983.

[EC90] G. Elber and E. Cohen. Hidden Curve Removal for Free Form Surfaces. In *Computer Graphics (Proc. SIGGRAPH)*, 1990.

[EG01] J. Elder and R. Goldberg. Image editing in the contour domain. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(3), 2001.

[EHW97] P. Eades, M. E. Houle, and R. Webber. Finding the best viewpoints for three-dimensional graph drawings. *Lecture Notes in Computer Science*, 1353:87–??, 1997.

[Elb95] G. Elber. Line art rendering via a coverage of isoparametric curves. *IEEE Trans. on Visualization and Computer Graphics*, 1(3):231–239, September 1995.

[Elk94] J. Elkins. *The Poetics of Perspective*. Cornell U. Pr., 1994.

[Fai98] Fairchild. *Color Appearance Models*. Addison-Wesley, 1998.

[Fer99] J. Ferwerda. Varieties of realism. talk, Cornell Workshop on Rendering, Perception, and Measurement, http://www.graphics.cornell.edu/workshop/, 1999.

[FLR+95] O. Faugeras, S. Laveau, L. Robert, G. Csurka, and C. Zeller. 3-d reconstruction of urban scenes from sequences of images. In A. Gruen, O. Kuebler, and P. Agouris, editors, *Automatic Extraction of Man-Made Objects from Aerial and Space Images*. Birkhauser, 1995.

[FS94] A. Finkelstein and D. Salesin. Multiresolution curves. In *Proc. SIGGRAPH*, 1994.

[GDCV98] J. Gomes, L. Darsa, B. Costa, and L. Velho. *Warping And Morphing Of Graphical Objects*. Morgan Kaufman, 1998.

[GFMS95] A. Glassner, K. Fishkin, D. Marimont, and M. Stone. Device-directed rendering. *ACM Trans. on Graphics*, 14(1):58–76, January 1995.

[GG99] G. Glaeser and E. Gröller. Fast generation of curved perspective for ultra-wide-angle lenses in vr applications. *Visual Computer*, 15:365–376, 1999.

[GG01] Gooch and Gooch. *Non-Photorealistic Rendering*. AK-Peters, 2001.

[GGSC98] A. Gooch, B. Gooch, P. Shirley, and E. Cohen. A non-photorealistic lighting model for automatic technical illustration. *Proc. SIGGRAPH*, 1998.

[GH00] R. Gershbein and P. Hanrahan. A fast relighting engine for interactive cinematic lighting design. In *Proc. SIGGRAPH*, Computer Graphics, 2000.

[GKR95] B. Guenter, T. Knoblock, and E. Ruf. Specializing shaders. *Proc. SIGGRAPH*, 1995.

[Gla00] Andrew Glassner. Cubism and cameras: Free-form optics for computer graphics. Technical Report MSR-TR-2000-05, Microsoft Research, 2000.

[GM97] J. Geigel and K. Musgrave. A model for simulating the photographic development process on digital images. *Proc. SIGGRAPH*, 1997.

[Gom56] Gombrich. *Art and Illusion*. Princeton Press, 1956.

[Gom82] E.H. Gombrich. *The image and the eye : further studies in the psychology of pictorial representation*. Cornell U. Pr., 1982.

[Goo98] A. Gooch. Interactive non-photorealistic technical illustration. Master's thesis, U. of Utah, 1998.

[Gre99] S. Green. Beyond photorealism. In *Eurographics Workshop on Rendering*, 1999.

[GRMS01] B. Gooch, E. Reinhard, C. Moulding, and P. Shirley. Artistic composition for image creation. In *Eurographics Workshop on Rendering*, 2001.

[Grö94] Eduard Gröller. Nonlinear ray tracing: visualizing strange worlds. *Visual Computer*, 11(5), 1994.

[GSG+99] B. Gooch, P. Sloan, A. Gooch, P. Shirley, and R. Riesenfeld. Interactive Technical Illustration. *ACM Symposium on Interactive 3D Graphics*, 1999.

[GSS+99] S. Green, D. Salesin, S. Schofield, A. Hertzmann, P. Litwinowicz, A. Gooch, C. Curtis, and B. Gooch. Non-Photorealistic Rendering. *SIGGRAPH Non-Photorealistic Rendering Course Notes*, 1999.

[Guz71] A. Guzman. Analysis of curved line drawings using context and global information. In B. Meltzer D. Mitchie, editor, *Machine Intelligence V6*, pages 325–375. Edinburgh U. Pr., 1971.

[GW92] M. Gleicher and A. Witkin. Through-the-lens camera control. *Computer Graphics (Proc. SIGGRAPH)*, 1992.

[HAA97] Y. Horry, K. Anjyo, and K. Arai. Tour into the picture: Using a spidery mesh interface to make animation from a single image. In *Proc. of SIGGRAPH 97*, 1997.

[Hae90] P. Haeberli. Paint by numbers: Abstract image representations. *Proce. SIGGRAPH*, 1990.

[Hag86] M. Hagen. *Varieties of Realism: Geometries of Representational Art*. Cambridge U. Pr., 1986.

[Hau01] A. Hausner. Simulating decorative mosaics. *Proc. SIGGRAPH*, 2001.

[HCS96] L. He, M. Cohen, and D. Salesin. The virtual cinematographer: A paradigm for automatic real-time camera control and directing. *Computer Graphics (Proc. SIGGRAPH)*, 1996.

[Her98] A. Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. *Proc. SIGGRAPH*, 1998.

[Her01] A. Hertzmann. Paint by relaxation. In *CGI*, 2001.

[HH90] P. Hanrahan and P. Haeberli. Direct WYSIWYG painting and texturing on 3D shapes. *Computer Graphics (Proc. SIGGRAPH)*, 1990.

[Hoc01] D. Hockney. *Secret Knowledge: Rediscovering the Lost Techniques of the Old Masters*. Viking Press, 2001.

[Hog81] B. Hogarth. *Dynamic Light and Shade*. Watson Guptill, 1981.

[HOT98] J. Hodgins, J. O'Brien, and J. Tumblin. Perception of human motion with different geometric models. *IEEE TVCG*, 4(4), October 1998.

[HP00] A. Hertzmann and K. Perlin. Painterly rendering for video and interaction. In *Non-Photorealistic Animation and Rendering*, 2000.

[Huf71] D. Huffman. Impossible objects as nonsense sentences. In B. Meltzer and D. Michie, editors, *Machine Intelligence 6*, pages 295–324. Edinburgh U. Pr., 1971.

[Hun95] Hunt. *The reproduction of Color (5th ed.)*. Kings Langley:Fountain Press, 1995.

[IMT99] T. Igarashi, S. Matsuoka, and H. Tanaka. Teddy: A sketching interface for 3d freeform design. *Proc. SIGGRAPH*, 1999.

[Kaj86] J. Kajiya. The rendering equation. In *Computer Graphics (Proc. SIGGRAPH)*, 1986.

[Kem90] M. Kemp. *The Science of Art*. Yale University Press, 1990.

[KMN+99] M. Kowalski, L. Markosian, J. D. Northrup, L. Bourdev, R. Barzel, L. Holden, and J. Hughes. Art-based rendering of fur, grass, and trees. *Proc. SIGGRAPH*, 1999.

[Koe90] J. Koenderink. *Solid Shape*. MIT Press, 1990.

[KS00] C. Kaplan and D. Salesin. Escherization. *Proc. SIGGRAPH*, 2000.

[Kub86] Michael Kubovy. *The Psychology of Perspective and Renissance Art*. Cambridge University Press, 1986.

[Lan77] E. H. Land. The retinex theory of color vision. *Scientific American*, 237(6):108–128, December 1977.

[Lev98] Jonathan Levene. A framework for non-realistic projections. Master's thesis, Massachusetts Institute of Technology, 1998.

[LG96] H. Löffelmann and E. Gröller. Ray tracing with extended cameras. *J. of Visualization and Comp. Animation*, 7(4):211–227, 1996.

[Lit91] P. Litwinowicz. Inkwell: A 2 -d animation system. *Computer Graphics (Proc. SIGGRAPH)*, 1991.

[Lit97] P. Litwinowicz. Processing images and video for an impressionist effect. *Proc. SIGGRAPH*, 1997.

[LS95] J. Lansdown and S. Schofield. Expressive Rendering: A Review of Nonphotorealistic Techniques. *IEEE CG & A*, 15(3):29–37, 1995.

[MAB+97] J. Marks, B. Andalman, P. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, and S. Shieber. Design galleries: A general approach to setting parameters for computer graphics and animation. In *Proc. SIGGRAPH 97*, 1997.

[Max83] Nelson L. Max. Computer graphics distortion for IMAX and OMNIMAX projection. In *Nicograph '83 Proceedings*, pages 137–159, December 1983.

[Mei96] B. Meier. Painterly rendering for animation. *Proc. SIGGRAPH*, 1996.

[MGT00] D. Martin, S. Garcia, and J. C. Torres. Observer dependent deformations in illustration. In *Non-Photorealistic Animation and Rendering*, 2000.

[Mil91] Millerson. *Lighting for Television and Film, 3rd ed.* Focal Press, 1991.

[MKT+97] L. Markosian, M. Kowalski, S. Trychin, L. Bourdev, D. Goldstein, and J. Hughes. Real-time nonphotorealistic rendering. *Proc. SIGGRAPH*, 1997.

[MMK+00] L. Markosian, B. Meier, M. Kowalski, L. Holden, J.D. Northrup, and J. Hughes. Art-based rendering with continuous levels of detail. In *Non-Photorealistic Animation and Rendering*, 2000.

[NPA00] NPAR. Discussion. First International Symposium on Non Photorealistic Animation and Rendering, 2000.

[OCDD01] B. M. Oh, M. Chen, J. Dorsey, and F. Durand. Image-based modeling and photo editing. *Proc. SIGGRAPH*, 2001.

[OH95] V. Ostromoukhov and R. Hersch. Artistic screening. *Proc. SIGGRAPH*, 1995.

[Ost99] V. Ostromoukhov. Digital facial engraving. *Proc. SIGGRAPH*, 1999.

[Pan27] Erwin Panofsky. *Perspective As Symbolic Form (La perspective comme forme symbolique)*. Zone Books, 1927.

[Par01] R. Parent. *Computer Animation: Algorithms and Techniques,*. Morgan-Kaufmann, 2001.

[PF92] P. Poulin and A. Fournier. Lights from highlights and shadows. In *ACM Symposium on Interactive 3D Graphics*, 1992.

[PFFL00] L. Petrovic, B. Fujito, A. Finkelstein, and L.Williams. Shadows for cel animation. *Proc. SIGGRAPH*, 2000.

[PFG00] F. Pellacini, J. Ferwerda, and D. Greenberg. Toward a psychophysically-based light reflection model for image synthesis. *Proc. SIGGRAPH*, 2000.

[PHM90] D. Pearson, E. Hanna, and K. Martinez. Computer generated cartoons. In Barlow, Blakemore, and Weston-Smith, editors, *Images and Understanding*. Cambrdge U. P., 1990.

[PHMF01] E. Praun, H. Hoppe, M., and A. Finkelstein. Real-time hatching. *Proc. SIGGRAPH*, 2001.

[Pir70] M. Pirenne. *Optics, Painting and Photography*. Cambridge U. Pr., 1970.

[PRJ97] P. Poulin, K. Ratib, and M. Jacques. Sketching shadows and highlights to position lights. In *Proc. Computer Graphics International*, 1997.

[Rad99] P. Rademacher. View-dependent geometry. *Proc. SIGGRAPH*, 1999.

[RAGS01] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer Graphics and Applications*, 21(4):34–41, 2001.

[RB98] P. Rademacher and G. Bishop. Multiple-center-of-projection images. In *Proc. SIGGRAPH*, 1998.

[RC99] R. Raskar and M. Cohen. Image Precision Silhouette Edges. In *Proc. 1999 ACM Symp. on Interactive 3D Graphics*, 1999.

[Rey00] C. Reynolds. Stylized depiction in computer graphics non-photorealistic, painterly and 'toon rendering. http://www.red3d.com/cwr/npr/, 2000.

[RH99] V.S. Ramachandran and W. Hirstein. The science of art: A neurological theory of aesthetic experience. *Journal of Consciousness Studies*, 6(6/7), 1999.

[RK00] C. Rössl and L. Kobbelt. Line-art rendering of 3d models, 2000.

[RS56] T. Ryan and C. Schwartz. Speed of perception as a function of mode of representation. *Am. J. of Psychology*, 69:60–69, 1956.

[SABS94] Michael P. Salisbury, Sean E. Anderson, Ronen Barzel, and David H. Salesin. Interactive pen-and-ink illustration. *Proceedings of SIGGRAPH 94*, pages 101–108, July 1994. ISBN 0-89791-667-0. Held in Orlando, Florida.

[SB99] M. Sousa and J. Buchanan. Observational model of blenders and erasers in computer-generated pencil rendering. In *Graphics Interface*, 1999.

[SD96] S. Seitz and C. Dyer. View morphing: Synthesizing 3d metamorphoses using image transforms. *Proc. SIGGRAPH*, 1996.

[SDS+93] C. Schoeneman, J. Dorsey, B. Smits, J. Arvo, and D. Greenberg. Painting with light. *Computer Graphics*, 1993.

[SF91] Dorée Duncan Seligmann and Steven Feiner. Automated generation of intent-based 3d illustrations. *Computer Graphics (Proceedings of SIGGRAPH 91)*, 25(4):123–132, July 1991. ISBN 0-201-56291-X. Held in Las Vegas, Nevada.

[SGG+00] P. Sander, X. Gu, S. Gortler, H. Hoppe, and J. Snyder. Silhouette clipping. *Proc. SIGGRAPH*, 2000.

[SK98] S. Seitz and K. Kutulakos. Plenoptic image editing. In *Proc. 5th Int. Conf.on Computer Vision*, 1998.

[SL01] R. Shacked and D. Lischinski. Automatic lighting design using a perceptual quality metric. *Computer Graphics Forum*, 20(3):215–226, 2001.

[SMGG01] P. Sloan, W. Martin, A. Gooch, and B. Gooch. The lit sphere: A model for capturing npr shading from art. In *Proc. Graphics Interface*, 2001.

[SSS74] Ivan E. Sutherland, Robert F. Sproull, and Robert A. Schumacker. A characterization of ten hidden-surface algorithms. *ACM Computing Surveys*, 6(1):1–55, March 1974.

[SST89] Saito, Shinya, and Takahashi. Highlighting rounded edges. In *Computer Graphics International*, 1989.

[ST90] T. Saito and T. Takahashi. Comprehensible Rendering of 3-D Shapes. In *Computer Graphics (Proc. SIGGRAPH)*, 1990.

[SWHS97] M. Salisbury, M. Wong, J. Hughes, and D. Salesin. Orientable textures for image-based pen-and-ink illustration. *Proc. SIGGRAPH*, 1997.

[TB96] G. Turk and D. Banks. Image-guided streamline placement. In *Proc. SIGGRAPH*, 1996.

[TDM01] O. Tolba, J. Dorsey, and L. McMillan. A projective drawing system. In *ACM Symp. on Interactive 3D Graphics*, 2001.

[TFN99] Saeko Takagi, Issei Fujishiro, and Masayuki Nakajima. Volumetric modeling of artistic techniques in colored pencil drawing. In *SIGGRAPH: Conf. abstracts and applications*, 1999.

[TJ81] F. Thomas and O. Johnston. *The Illusion of Life : Disney Animation*. Abbeville Pr., 1981.

[TT99] Tumblin and Turk. LCIS: a boundary hierarchy for detail-preserving contrast reduction. In *Proc. Siggraph*, 1999.

[TTT91] Tanaka, Tokiichiro, and Takahashi. Precise rendering method for edge highlighting. In *Computer Graphics International*, 1991.

[Tum99] Jack Tumblin. *Three methods of detail-preserving contrast reduction for displayed images*. PhD thesis, College of Computing Georgia Inst. of Technology, September 1999.

[Uli87] R. Ulichney. *Digital Halftoning*. MIT Press, 1987.

[VB99] O. Veryovka and J. Buchanan. Halftoning with image-based dither screens. *Graphics Interface*, 1999.

[vH81] Hermann von Helmoholtz. On the relation of optics to painting. In *Popular scientific lectures*. Appleton, 1881.

[Web83] Webster. *Ninth New Collegiate Dictionary*. Merriam Webster, 1983.

[WFH+97] D. Wood, A. Finkelstein, J. Hughes, C. Thayer, and D. Salesin. Multiperspective panoramas for cel animation. *Proc. SIGGRAPH*, 1997.

[Wil90] J. Willats. The draughtsman's contract. In H. Barlow, C. Blakemore, and M. Weston-Smith, editors, *Images and Understanding*. Cambrdge U. Pr., 1990.

[Wil91] L. Williams. Shading in two dimensions. In *Proc. of Graphics Interface '91*, pages 143–151, June 1991.

[Wil97] J. Willats. *Art and Representation*. Princeton U. Pr., 1997.

[WS94] G. Winkenbach and D. Salesin. Computer-generated pen-and-ink illustration. *Proc. SIGGRAPH*, 1994.

[ZB95] D. Zorin and A. Barr. Correction of geometric perceptual distortion in pictures. In *Proc. SIGGRAPH*, 1995.

[ZBr] Pixelogic ZBrush. http://pixologic.com/.

[ZDPSS01] L. Zhang, G. Dugas-Phocion, J. Samson, and S. Seitz. Single view modeling of free-form scenes. In *Proc. of CVPR*, 2001.

[Zek00] S. Zeki. *Inner Vision : An Exploration of Art and the Brain*. Oxford U. Pr., 2000.

[ZH00] D. Zorin and A. Hertzmann. Illustrating smooth surfaces. *Proc. SIGGRAPH*, 2000.

[ZHH96] R. Zeleznik, K. Herndon, and J. Hughes. SKETCH: An interface for sketching 3D scenes. *Proc. SIGGRAPH*, 1996.