User-Guided Composition Effects for Art-Based Rendering

Michael A. Kowalski †∗ *John F. Hughes* † *Cynthia Beth Rubin* ‡ *Jun Ohya* §∗

Brown University Rhode Island School of Design Waseda University

Abstract

We apply simple techniques from traditional artistic composition to the art-based rendering of interactive 3D scenes. A human scene-modeler makes choices about composition in a scene and our system dynamically adjusts the rendering attributes of objects in the scene to achieve the desired effects for a given view. We can selectively group scene elements through shared tone, color, and outline, so as to simplify and structure an image. This can be used, together with controlled level of detail, to emphasize important objects. Finally, we show a technique for adaptively changing color or other attributes to control the contrast of adjacent elements in the picture. We also briefly discuss ideas about larger-scale compositional issues.

CR Categories and Subject Descriptors: I.3.3 [Computer Graphics]: Picture/Image Generation - Display algorithms.

Additional Key Words: Non-photorealistic rendering, Procedural textures.

1 Introduction

Much of the work in non-photorealistic rendering (NPR) has concentrated on such issues as the production and arrangement of strokes. But artists do much more than this: they also choose subject matter, choose a position from which to paint it, and sometimes paint things in positions they never occupied, so as to provide a better balance on the canvas. This choice of how things should appear on the canvas – the 2D layout of regions of darkness and light, the determination of what to emphasize, the relative positions of the geometries of the shapes in an image – together constitute the *composition* of a painting or drawing.

Composition is aesthetically important, but there are probably underlying psychophysical explanations for some compositional techniques. Counterchange (described in Section 6.3), for example, may well have a basis in the physiology of the eye, and the nonlinearity of its response to incoming light. Such perceptual issues in rendering have been given considerable attention, e.g. [2, 4].

∗The authors developed this work in part while at ATR Media Integration & Communications Research Laboratories, Kyoto 619-0288, Japan.

†Department of Computer Science, Box 1910, Brown University, Providence, RI 02912. {mak,jfh}@cs.brown.edu

‡cbrubin@risd.edu

§Global Information and Telecommunication Institute, Waseda University, Tokyo 169-0051, Japan. ohya@giti.waseda.ac.jp

This paper presents some preliminary ideas and techniques for adjusting *rendering* parameters based on compositional needs in an interactive scene or animation. In particular, scene modelers can specify high-level parameters such as “emphasis,” which our system interprets according to rules (such as “object grouping rules” or relative placement), to yield parameters that influence low-level rendering techniques.

The specific effects we demonstrate are rendering the background (or other de-emphasized objects) with decreased tone/color variation, drawing distant groups of objects with merged outlines, and shifting the foreground object tone to contrast with the background.

These are only a few of the many tools constituting composition. The larger issues of 2D design and layout in an image, the choice of view or placement of objects — these we leave for future work. But as many artists have noted, composition rules are useful, but the art of composition is in knowing which rules to apply and which to ignore. Because of this, we feel the most promising current approach is a “mixed initiative” one in which a scene modeler declares certain compositional goals for a scene (e.g., which objects are important in a scene, where does contrast matter, etc.), places a virtual camera, and then lets the computer perform minor alterations to improve composition. In this paper, those alterations are made through the modification of rendering attributes.

2 Related work

Many art instruction books discuss composition in depth [3, 13, 15], and Calahan has written a nice introduction to composition ideas and techniques for those in computer graphics [1]. But composition has attracted only a little attention in graphics.

In the context of providing technical illustrations with insets, etc., Seligman and Feiner [14] described a system that automatically chooses objects to include, rendering, viewing and lighting parameters, and the layout of images in a composite illustration. In the area of non-photorealistic rendering, Strothotte *et al.* [16] pointed out the importance of the choice of rendering style on a viewer’s interpretation of a scene, and in directing the viewer’s attention. Masuch *et al.* [11] developed a system for rendering off-line animations using stylized lines and showed how changing line styles over time can be used to emphasize certain elements. The stylistic changes are specified by keyframes supplied to the renderer. In the interactive rendering of 3D models, Gooch *et al.* [6] apply techniques from traditional technical illustration to emphasize the important structure and detail of an object, but they do not specifically address such issues in rendering multiple objects in the context of a scene.

There has been some work addressing the view-dependent control of detail in NPR (for example, the procedural stroke textures developed by Winkenbach and Salesin [17] and the work of Markosian *et al.* on graftal textures [7, 9]). But in these systems, level of detail is typically determined *locally* for each object or surface rather than supporting overall compositional intent.

3 Target effects and approach

Artists often simplify regions in a picture to draw attention to other more important elements, and for expressive and aesthetic reasons. One technique is to reduce *detail* in unimportant areas. But artists can also simplify in subtle ways by reducing the variation in *tone* and *color* [3, 15]. For example, in a painting of a foreground figure in a garden, the background trees, which contain many shades of green, may be painted in a single "average" color to help reduce their prominence.

Simplification can be achieved through line as well as color. For example, a distant mountainous horizon may be drawn with one stroke, even though it consists of multiple forms that happen to overlap in the view. In a sense, for the purpose of drawing such a line, these objects are treated as a single entity. This conveys the essential boundary between the large land masses and sky with a single economical sweep, omitting the less significant boundaries between smaller or less visible forms.

Another way to bring out a foreground element is to lighten the color of the foreground object, making it contrast with a dark background; this slight alteration of the color of an object to make it contrast with the background is called "counterchange" [3, 13].

In the remainder of this paper, we present data structures and techniques that support compositional rendering choices and a control mechanism for making such choices automatically under certain conditions. In particular, we introduce a *composition manager* that controls rendering of multiple objects on a global level. We also let the scene designer specify the *visual grouping* of objects in the scene. Such groupings allow simplification effects wherein multiple objects share visual attributes (such as tone, color, or outline) as a function of compositional parameters.

4 Software framework

We build on the work of Markosian *et al.* [7, 10, 9]. Their framework provides for the rendering of polyhedral models using procedural textures (*textures* for short) that support several stylistic effects. In this framework, a scene modeler can divide the surface of a model into subregions called *patches*. She can then assign to each patch a texture that renders the patch in a particular style. Supported styles range from simple outline renderings to stroke-based textures that suggest the appearance of complex natural detail.

The Markosian *et al.* system produces, for each frame, an *ID reference image*, containing the object IDs that uniquely identify the patch that is rendered at each pixel location. (In our terminology, we say that ID pixels *belong* to a patch if that patch's ID appears at those pixel locations in the ID image.) As we discuss in Section 5.1, such ID information is important in our application for determining screen-space adjacency of scene elements.

4.1 Composition manager

To orchestrate the rendering of various objects in the scene we have incorporated a *composition manager* into our implementation of the Markosian *et al.* system. This manager analyzes each frame and provides guidance to each "texture" on how it should render itself to meet the particular visual qualities required by the composition in the current view. For example, the composition manager might indicate that a group of objects should render with reduced detail or emphasis, to draw attention to a different part of the scene.

The interface between the composition manager and the textures is abstract whenever possible. The compositional guidance given by the manager is often on a high level and avoids specifying exact rendering attributes and behaviors. For example, to indicate how prominent a texture should be, the manager sets a single parameter, "emphasis," that is between 0 and 1. The texture's implementation determines how this value is interpreted. Thus the same variable can control the appearance of more than one kind of texture: for a texture that generates grass, the level of emphasis may affect the density of the tufts; for an outline texture the emphasis may be reflected in the weight or thickness of the outlines. Making the interface abstract also provides flexibility: for a given composition manager, textures of somewhat different styles may be interchanged while maintaining similar compositional behaviors and effects.

With this approach, the modeler controls the composition of her scene by (1) specifying rules that the manager follows to guide the rendering and (2) designing textures that interpret such guidance.

4.2 Composition-related data structures

For visual grouping the composition manager maintains *compositional group* data structures that contain not only the modeler-specified list of the objects or patches in the group, but also a description of the visual attributes (such as *group color* or *group linestyle*) associated with the group. In the case of color, each object in the group may render either with its own individual color or, when compositional goals demand that grouping be fully in effect, with the group color. The composition manager computes a *degree of grouping* parameter g for the group. Each object within the group uses this to linearly interpolate between its individual base color C_i and the group color C_g to yield color $C = (1 - g)C_i + gC_g$.

To make decisions based on an object's image-space context, as we must for counterchange, for example (Section 6.3), we use an *image-region* data structure and related procedures. This structure provides information about the configuration of a 3D object as a screen-space entity, including information about the screen-space boundaries of a given patch. Such boundaries are defined as the collection of ID pixels belonging to the patch that are adjacent to at least one pixel where the patch does not appear. To identify such boundary pixels, we iterate once over all the pixels belonging to a patch P in the current frame. For each such pixel location p, we check for any pixel o immediately adjacent to p in the ID reference image which contains the ID of some patch P_o other than P. The color assigned to P_o can then be be queried and recorded. When all the boundary pixels have been found, the average color along the border of patch P can be computed.

5 Rendering techniques

In this section we describe our adaptations of existing art-based rendering algorithms to convey compositional effects.

5.1 Group outline rendering

To allow simplified representation of the outlines of overlapping forms, we built a texture that renders *group outlines*. For a polyhedral model, the outline or silhouette consists of the visible portions of the model's *silhouette edges.* (A silhouette edge is one that joins a front-facing and a back-facing triangle.) We define the *group outline* to be the subset of silhouettes that are on the *group's* image-space boundary. (The rendering of the mountains in Figure 2(c) shows the group outlines generated by our algorithm. These are a subset of all the outlines, shown in Figure 2(d).)

To identify this subset of outlines, we use grouping information and a modified version of the silhouette-rendering algorithm of Northrup *et al.* [12]. This algorithm first identifies the silhouette edges, and then builds stroke paths from the image-space projections of the visible segments of these edges.

We modify the first step: rather than considering *all* silhouette edges, we apply the algorithm only to the silhouette edges that make up the group outline. These can be easily identified via the ID reference image: we render all silhouette edges and all faces, each in a different color, in the reference image; from this the list of pixel locations of all the silhouette edges can be found in constant time. If an edge location p is adjacent to at least one pixel o not belonging to the group, and if the surface visible at o is at a greater depth than p, then the edge at p is a group outline edge. From these edges, the Northrup algorithm provides stroke paths for rendering.

5.2 Generating detail

To control detail in the context of composition, we adapt the *graftals* described by Markosian *et al.* [9]. Their graftals are elements, distributed on the surface of the model, that can generate geometry and strokes as needed in the current view to suggest the appearance of complex detail such as vegetation. The amount of detail generated (e.g., the number of blades of grass) is controlled by a *level of detail* parameter, λ, that is computed by an individual graftal or by a collection of graftals. λ is determined in a view-dependent way, usually in part as a function of the graftal's distance from the camera. Thus, for example, when a graftal is far away it may not draw at all, but as it nears the camera more detail is gradually introduced. We modify this behavior by scaling λ by the current level of emphasis e that the composition manager has set for the patch containing the graftal. The perceived detail is then attenuated both by geometric considerations, such as distance, and by the graftal's compositional importance.

5.3 Shading model

For our shading effects we use an approximation of the non-photorealistic lighting model proposed by Gooch *et al.* [5]. Their approach utilizes change in hue as well as luminance to convey surface orientation. An advantage of this method is that it works well in conjunction with line drawing, avoiding extreme darks in the shading that might obscure the outlines. Using this scheme, we illuminate the model with an ambient light and two opposing directional lights, orthogonal to the viewing direction. We then set the colors of the lights so as to create a warm-to-cool transition of hue across the surface[1]. We use this to illuminate the rocks in the scene shown in Figure 1. To diminish tonal contrast in the shading, we reduce the intensity of the directional lights, but maintain the ambient illumination. This gives a flatter, more uniform tone to an entire object when it is to be de-emphasized.

For the black-and-white tree in Figure 3, we use the same arrangement of lights as proposed by Gooch *et al.*, but depart from their color scheme, using gray for both the ambient illumination and the upper light. The lower light has equal negative values in all color channels, adding a dark rim along the lower surface of the model.

6 Compositional rules: implementation and results

Using the composition manager, its grouping data structures, and the rendering techniques described above, one can attain several specific compositional effects. Here we give three examples.

[1]We use RGB colors (.5,.5,.5), (.5, .5, 0), (-.5, -.5, 0) for the ambient, upper, and lower lights, respectively.

6.1 Importance hierarchy

In our first example we show how an importance hierarchy can be indicated through color grouping, reduced contrast, and attenuation of detail, and how the transition of the background elements into grouped and de-emphasized states can be achieved smoothly. We do this for the simplest possible hierarchy: a foreground object that is important, and a background that is not. In general, though, one can create a hierarchy in which an emphasis e of an object at level i multiplies the emphasis of all lower-level objects by $1 - e$.

In Figures 1(a)-(d), the dinosaur is designated as the important scene element. All other parts of the scene (the stylized rocks and the land) are secondary. In addition, the rocks are organized into a group structure and are assigned a group color (we use the average of all the individual colors). For maximum unifying effect, the textures assigned to the rocks are designed to reduce their *shading contrast* (see Section 5.3) and the weight of their outlines[2] when grouping is in effect, in addition to the transition to the group color. Because we want to simplify the background most when the dinosaur is the most prominent, the degree of grouping specified for the rocks is directly proportional to the dinosaur's level of emphasis.

The level of emphasis for the dinosaur is based on an estimate of its current screen size and on a measure of adjacency to the screen center. If the object is above a specified screen size and within a specified distance from the center of the screen, it is fully emphasized, and its emphasis is gradually scaled down if it falls outside these thresholds.

In Figure 1(a) the dinosaur is not prominent and thus has no effect on the rocks and the grass, which render with full emphasis. As the dinosaur is moved to a more visible position closer to the screen center ((b) and (c)), the composition manager automatically begins to de-emphasize the background elements. The most distant grass blades begin to fade. The rocks collectively begin to shift from their individual colors towards their single group color. The shading contrast in the rocks is noticeably reduced. As the camera zooms in, so that the dinosaur occupies most of the view (d), the rendering of the background elements is suppressed to the maximum degree. Individual grass blades are gone, and the rocks all render in a single group color with faded outlines and flatter shading. Figure 1(e) shows the last view with the composition manager disabled, hence no background suppression.

6.2 Grouping of distant objects

Grouping naturally helps convey distance; even Leonardo recommended that painters de-emphasize distant boundaries. We try to follow this advice in the next example, grouping elements based on distance by fading intra-group object boundaries. The three mountains shown in Figure 2 (each of which has been assigned a group-outline texture) form a single group. The goal of the composition manager is to merge the individual mountains into a shape defined by a single outline when they are sufficiently distant.

We group objects based on an approximation of the screen-size of the group, namely the screen size of the group's bounding sphere. Using this the composition manager begins to affect grouping once the size falls below a modeler-specified threshold. We also let the scene modeler define the length of the interval over which the transition should take place. Guided by the composition manager, all group members change from individual to fully grouped mode at the same rate.

[2]In this example, we draw the outlines by simply rendering the mesh's silhouette edges as OpenGL line strips, finding such edges using the randomized algorithm by Markosian *et al.* [8].

Figure 2(a) shows the group at a close distance, where grouping has not yet taken effect. Every mountain draws all of its outlines. In (b), the camera has moved back and the initial effects of the grouping can be perceived. The interior boundaries of the group are beginning to fade away. Finally, in part (c), grouping is completed. All interior outlines have disappeared, leaving only the exterior silhouette of the mountains. Part (d) shows the result without grouping.

6.3 Adaptive contrast enhancement

In our last example, we demonstrate *counterchange*, a compositional technique based on contrast enhancement, in which we shift the tone or color of a foreground element so as to contrast with the background, making the foreground element stand out. This effect seems to work best in pictures where the background consists of large areas of simplified tone or color, and hence it might naturally be combined with the grouping of background elements.

Figure 3 shows our implementation of a procedural texture that produces this effect. We've used black-and-white to emphasize the effect. The counterchange texture makes use of the image region data structure, described in Section 4.2, to determine the average adjacent background color along its image space boundaries. Based on the current background color, the counterchange texture modifies its own color and shading to enhance contrast with the background. The scene modeler specifies the amount by which both the color and lighting are permitted to change. For the counterchange texture assigned to the tree in Figure 3, we allow the color to vary between the graylevels 0.37 and 0.63. These values were deliberately chosen to be close to the range of background colors of 0.3 and 0.7.

In this example, we vary the color and lighting according to the following rules. Let C_{dark} and C_{light} represent the darkest and lightest colors, respectively, that the object may assume, and let C_{bg} represent the current average background color. Then the object's current color is given by C_{dark} if $C_{\text{bg}} \geq C_{\text{light}}$, by C_{light} if $C_{\text{bg}} \leq C_{\text{dark}}$, and by $C_{\text{dark}} + C_{\text{light}} - C_{\text{bg}}$ otherwise. In addition, the lighting is modified simultaneously to reinforce the color shift. (The lighting model for this example is described is Section 5.3.) Each light is interpolated simultaneously with the color, so that when the color is the brightest, the white light from above is at its maximum intensity and the negative light from below contributes nothing. Conversely, when the object color is darkest, the negative light is the most intense, thus reinforcing the darks, and the white light has 0 intensity.

6.4 Performance

The system runs at interactive rates for most of the effects we have shown. The dinosaur scene and the counterchange demonstration both run at approximately 12 frames per second (640×480 pixels) on our test machine: a 450 MHz CPU Sun Ultra 60 with Elite 3D graphics. The group-outlines scene with the mountains is not interactive, but runs at approximately 3 - 4 frames per second.

7 Conclusion and future work

Our results show some interesting possibilities but they are preliminary. Whole areas of composition — the 2D layout of masses of darkness and light, camera positioning, object positions, etc. — remain unaddressed. The video clearly shows that composition for changing scenes must be more complex than a sequence of appropriately composed individual pictures; at present we have no clear ideas on how to express time-aggregated compositional goals. The literature on cinematographic lighting is surely relevant.

We suspect composition in some well-defined domains (e.g., repair manuals) may be substantially automatable, so constant scene-designer involvement will no longer be essential. But we generally feel designer involvement is critical, because composition is an expressive tool. The challenge is to choose the appropriate dividing line between computer assistance and author initiative.

8 Acknowledgments

We thank Barb Meier, Lee Markosian, and David Laidlaw for valuable suggestions. Thanks also to Andy van Dam, the Brown Graphics Group, Ryohei Nakatsu of ATR Labs, and our sponsors who helped support this work: the NSF Science and Technology Center for Computer Graphics and Scientific Visualization, Adobe, Advanced Network and Services, Alias/Wavefront, Department of Energy, IBM, Intel, Microsoft, National Tele-Immersion Initiative, and Sun Microsystems.

References

[1] Sharon Calahan. Storytelling through lighting, a computer graphics perspective. In *Advanced RenderMan: Creating CGI for Motion Pictures*. M organ Kaufman, San Francisco, 2000.

[2] K. Chiu, M. Herf, P. Shirley, S. Swamy, C. Wang, and K. Zimmerman. Spatially nonuniform scaling functions for high contrast images. In *Proceedings of Graphics Interface 93*, pages 245–253, May 1993.

[3] Bernard Dunstan. *Composing Your Paintings*. Taplinger, 1979.

[4] J. A. Ferwerda, S. Pattanaik, P. Shirley, and D. P. Greenberg. A model of visual adaptation for realistic image synthesis. In *SIGGRAPH 96 Conference Proceedings*, pages 249–258. Addison Wesley, August 1996.

[5] A. Gooch, B. Gooch, P. Shirley, and E. Cohen. A non-photorealistic lighting model for automatic technical illustration. *Computer Graphics*, 32(Annual Conference Series):447–452, August 1998.

[6] B. Gooch, P.-P. J. Sloan, A. Gooch, P. Shirley, and R. Riesenfeld. Interactive technical illustration. *1999 ACM Symposium on Interactive 3D Graphics*, pages 31–38, April 1999.

[7] M. A. Kowalski, L. Markosian, J. D. Northrup, L. Bourdev, R. Barzel, L. S. Holden, and J. F. Hughes. Art-based rendering of fur, grass, and trees. *Computer Graphics*, 33(Annual Conference Series):433–438, 1999.

[8] L. Markosian, M. A. Kowalski, S. J. Trychin, L. D. Bourdev, D. Goldstein, and J. F. Hughes. Real-time nonphotorealistic rendering. *Computer Graphics*, 31(Annual Conference Series):415–420, August 1997.

[9] L. Markosian, B. J. Meier, M. A. Kowalski, L. S. Holden, J. D. Northrup, and J. F. Hughes. Art-based rendering with continuous levels of detail. *NPAR 2000: First International Symposium on Non-Photorealistic Animation and Rendering*, pages 59–66, June 2000.

[10] Lee Markosian. *Art-based Modeling and Rendering for Computer Graphics*. PhD thesis, Brown University, 1999.

[11] M. Masuch, L. Schuhmann, and S. Schlechtweg. Frame-to-frame-coherent line drawings for illustrated purposes. In *Proceedings of Simulation und Visualisierung '98*, 1998.

[12] J. D. Northrup and L. Markosian. Artistic silhouettes: A hybrid approach. *NPAR 2000: First International Symposium on Non-Photorealistic Animation and Rendering*, pages 31–38, June 2000.

[13] John Ruskin. *The Elements of Drawing*. The Herbert Press, 1991.

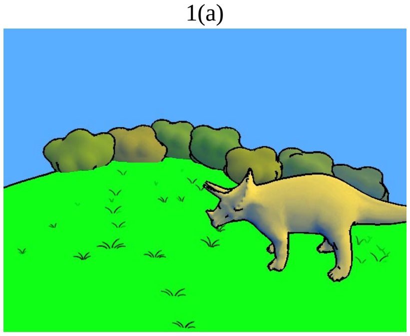
[14] D. Seligman and S. Feiner. Specifying composite illustrations with communicative goals. In *Proceedings of ACM UIST '89*, pages 1–9, 1989.

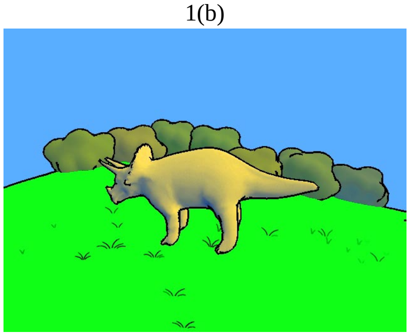
[15] Harold Speed. *Oil Painting Techniques and Materials*. Dover, 1987.

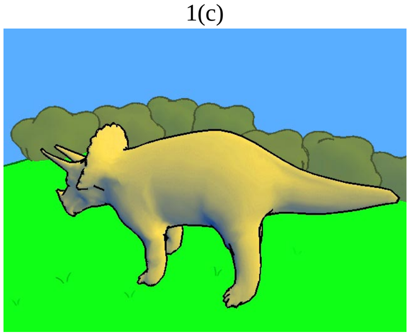
[16] T. Strothotte, B. Preim, A. Raab, J. Schuman, and D. Forsey. How to render frames and influence people. *Computer Graphics Forum*, 13(3):455–466, September 1994.

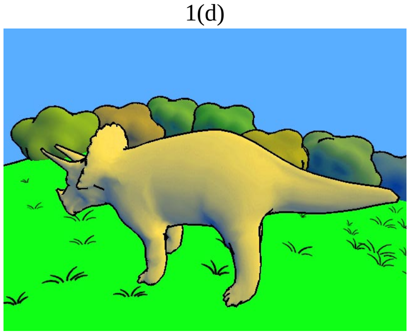
[17] Georges Winkenbach and David H. Salesin. Computer–generated pen–and–ink illustration. In *SIGGRAPH 94 Conference Proceedings*, pages 91–100. ACM SIGGRAPH, July 1994.

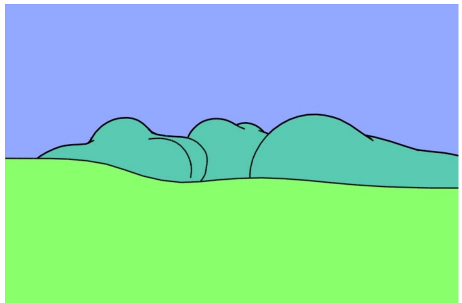

1(a)

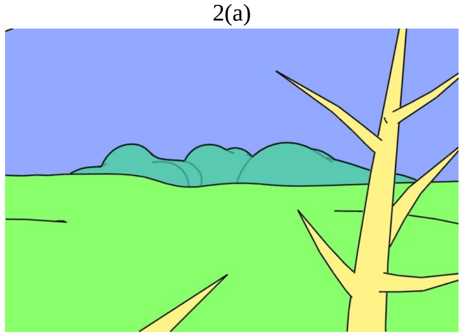

1(b)

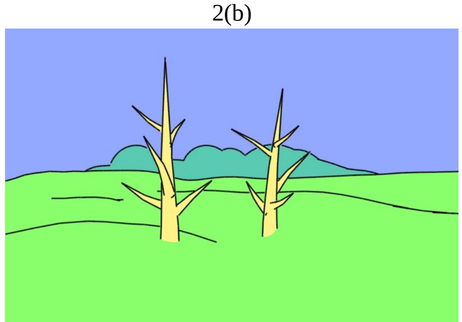

1(c)


1(d)


1(e) No grouping or suppression.


2(a)


2(b)


2(c)

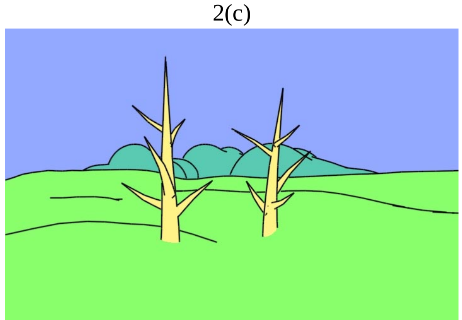
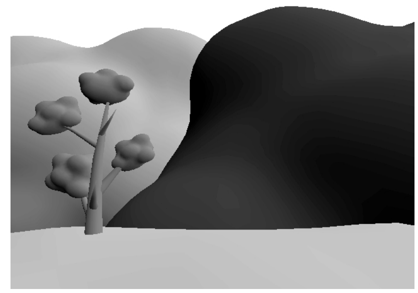
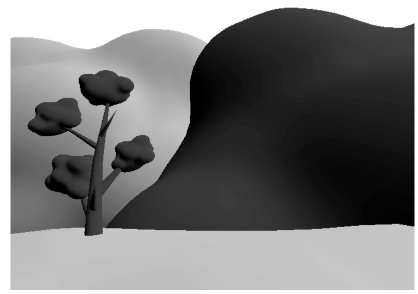
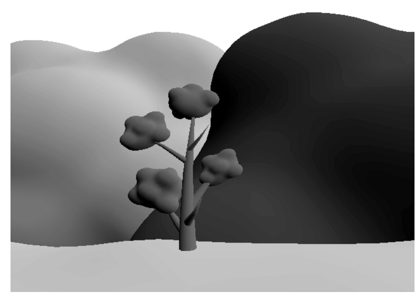
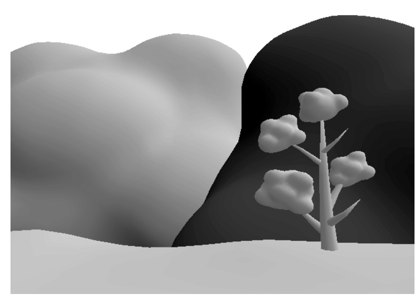

2(d) No grouping.

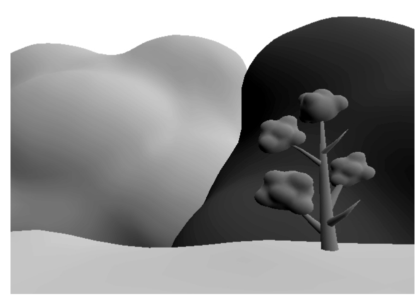
Figure 1. Simplification of the background. **Figure 2.** Grouping through outline. **Figure 3.** “Counterchange”. (b) and (d) To enhance contrast and definition, the tree automatically adjusts its color and shading to appear darker or lighter depending on the current background. (c) The tree is in the middle of a transition between darkened and lightened states. (a) and (e) The same views as in (b) and (d) but with no counterchange; the tree renders with constant color and shading in both images.


3(a) No counterchange.


3(b)


3(c)


3(d)


3(e) No counterchange.