Understanding Visual Effects in a Windowed Environment

Frankie K. Sun[1]
William B. Cowan
Kellogg S. Booth

Computer Graphics Laboratory
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1

Abstract

We describe five demonstration programs for displaying variants of effects reported in the vision literature. The demonstrations use a windowed workstation environment, the X Window System Version 11. The performance of the window system is analyzed in terms of our implementation experience and suggestions are made for future window system architectures based on that experience. The visual effects are then discussed in terms of what they teach us about the display of information within a windowed environment and the tools that the visual effects themselves provide for improving that environment.

Résumé

Nous décrivons cinq programmes de démonstration pour afficher les différents effets de la vision, tel que la décrit la documentation courante sur le sujet. Les démonstrations utilisent un environment de station de travail à fenêtres multiples, plus particulièrement le système X Window, version 11. Les performances y sont analysées à la lumière de notre propre expérience tout comme sont présentées nos suggestions pour améliorer les futures architectures des systèmes à fenêtres. Nous élaborons ensuite sur la nature de l'affichage de l'information dans un tel environnement pour conclure sur la pertinence des travaux sur les effets de la vision pour améliorer les systèmes à fenêtres multiples.

Keywords: animated images, colour, lookup table, visual effect, window system, X11.

Introduction

Much is known about visual effects that occur in everyday life. Such effects influence the way that information is perceived. They are often called "visual illusions" for this reason. These effects occur when the human visual system is operating at the limits of its performance [1, 18]. We believe that such effects influence the way that information is perceived on a computer display and that an understanding of the parameters of real-time human performance is necessary for knowing how information is perceived on a computer display. Five demonstrations representing common classes of visual effects have been implemented using Version 11 of the X Window System [11, 19, 21] (referred to as X11 in the following) as part of an on-going research program examining the parameters of real-time human performance. The demonstrations have been designed for two purposes:

(a) to examine the suitability of X11 for supporting interactive applications under the assumption that the visual effects being studied are representative of display tasks that occur in real applications and the assumption that those tasks pose a reasonable performance measure for X11; and

(b) to test the hypothesis that common visual effects do occur in windowed workstation environments, to test the hypothesis that the effects do influence the way that information is perceived, and to provide a testbed for analyzing techniques that ameliorate the consequences of those effects.

The work reported here is primarily concerned with the first objective, the second objective being the focus of continuing research which we expect to report in more depth in subsequent publications. Our preliminary results indicate that, as might be expected, visual effects do indeed occur in windowed workstation environments, those effects do influence the way users perceive displayed information, and there are techniques that can be effectively employed to reduce or even eliminate the visual effects under appropriate circumstances.

The widespread use of windowed environments for computer workstations has created a situation in which an understanding of visual effects is necessary to assess the quality of current techniques for displaying information. We have chosen X11 for our studies because it represents the most prominent style of window system in current use. The Macintosh style of windows [12] is, for our purposes, the same as that employed by X11. The style of NeWS has some, but not all, of the additional features that we believe to be necessary for effective use of windowed environments.

---

[1] Present address: IBM Canada Ltd., 51/407, 895 Don Mills Road, North York, Ontario, Canada M3C 1W3.

This paper discusses the motivation for our studies, providing a justification for analyzing known visual effects in a windowed workstation environment and examining some of the features common to existing window systems that affect the implementation of our demonstration programs. Of particular interest are obstacles to the display of animated images and the coordination of multiple windows shared among independent applications, which we see as inherent in X11 and other existing window systems.

We begin with a brief description of five visual effects reported in the vision literature and comment on how demonstrations of those effects were implemented under X11. This is followed by a discussion of problems we encountered in the implementations and recommendations for future window systems that might overcome the problems. We conclude with an analysis of how the particular visual effects are related to important issues involving the use of windowed environments for computer workstations.

Demonstrating Visual Effects Using X11

Five effects were chosen as representatives of common classes of visual effects that can be expected to play a role in the effective perception of information displayed on a computer workstation.

Visual effect #1 – black/white contrast

The phenomenon of *simultaneous contrast* has been well-studied in the vision literature and its importance for computer displays has been discussed by many authors including earlier studies reported at this conference [9]. The effect is achieved by viewing a target area having a neutral gray value surrounded by a larger area that is either black or white. If the surround is black, the neutral gray appears to be light, but if the surround is white, the same neutral gray appears to be dark. This phenomenon is commonly thought to be a result of lateral inhibition caused by center-surround receptive fields in the ganglion cells of the retina [14].

To demonstrate this in a workstation environment, an X11 window was created in which two large rectangular areas, one black and one white, were displayed side-by-side with a smaller rectangle displayed in the middle of each (see Figure 1). Using a graphical slider displayed in a control window and adjusted by mouse movement, the gray level for one of the inner rectangles could be adjusted from full black to full white. Our implementation is similar to that described by Goetz and employs the color lookup tables supported by X11 to provide an efficient way to change the intensity of both the inner rectangles and the surrounds in response to slider movement.

A button in the control window toggles the display mode between a *normal state* with black and white surrounds and a *comparison state* in which the two gray levels are used to fill their respective larger rectangular areas. The comparison state is used to assess the degree to which a user has been successful in matching the two gray levels.

As predicted, the color of the surround influences the perception of the inner gray levels so that in the normal state a user adjusts the levels so that the gray with the black surround is in fact darker (and hence appears the same as) the gray with the white surround. In the comparison state, this mismatch is easily detected. Conversely, an exact match is easily achieved in the comparison state, but the matching gray levels appear to be quite different when viewed in the normal state where the two surrounds affect the perceived gray levels.

Visual effect #2 – yellow/gray contrast

The black/white contrast just described is a visual effect involving the perception of luminance. A similar effect exists involving the perception of chromaticity (color) [1]. In this effect rectangular surrounds, one yellow the other grey, are displayed with two embedded "X" figures that join across the boundary of the surrounds (see Figure 2). The "X" figures share a common color that is a perceptually equal mixture of the yellow and the grey from the surrounds. As before, a mouse-controlled slider is used to adjust the relative luminance (but not the chromaticity) of the yellow and gray.

At the point where the two colors are perceptually equiluminous, a *reversed ground* effect occurs (this indicates that equiluminance has been achieved) and each "X" takes on the color of the opposite surround. This effect demonstrates that simultaneous contrast is not limited to luminance, but also exists for purely chromatic contrasts.

Visual effect #3 – the nulling of apparent motion

The third effect also relies on achieving equiluminance, but in this case the result is not just a perceptual confusion of color or luminance, but the illusion of motion. In this demonstration, a window is created with a sequence of squares displayed, each drawn using a different color index. By altering the color lookup table entries corresponding to those indices, the colors change with time according to the pattern shown in Figure 3. If the colors are chosen so that #1 and #3 are black with #2 and #4 white, then the squares are perceived as moving to the left if each of the four configurations is displayed in sequence.

If instead colors #3 and #4 are replaced by red and green respectively, the illusion of motion is altered. If the luminance of the red is chosen to be perceptually less than that of the green, the squares continue to be seen as moving to the left, so that the red serves the same role as black in the original configuration and green assumes the role of white. If, on the other hand, the luminance of red is perceptually greater than that of green, these roles are reversed and the squares are perceived to be moving to the right. At the point where the red and green colors are equiluminous, the apparent motion disappears and a static (though flickering) image is perceived [13].

To demonstrate this using X11, a single line of squares is drawn, each with one of four distinct color indices. The color values of black, white, red, and green assigned to the four indices are cyclically shifted in the lookup tables with the relative luminance of red and green under control of a slider. The rate at which the color indices are cycled is determined by a second slider.

Visual effect #4 – the peripheral pinwheel

This effect illustrates the difference between foveal vision and peripheral vision. A "pinwheel" was designed, again using color lookup table techniques, so that a cyclic shift of the values associated with the color indices induces the appearance of rotational motion in the pinwheel (see Figure 4). The inner portion of the pinwheel is drawn using only values that differ in luminance. The outer portion of the pinwheel is drawn using red-green values, so that the appearance of rotational motion is induced by both changes in luminance and changes in chromaticity. If the red and green are equiluminous, the perception of the outer rotation relies entirely on the chromaticity differences [6].

Using traditional (non-computer display) techniques, it is well known that at equiluminance the apparent motion of the outer portion of the pinwheel is perceived to lag behind that of the inner portion and even to stop entirely as long as subjects keep the outer portion only in their peripheral vision. The reason for this is that the perception of color is highly localized in the fovea, with significantly less color acuity in the periphery.

In our demonstration using X11, the observed effect was achieved, although only weakly. If the viewer's gaze is kept at the center of the pinwheel as it rotates, the outer portion does appear to slow down compared to the inner portion. We encountered problems with maintaining a uniform rate of rotation that probably interfered with the main effect. These problems are discussed later in the paper.

Visual effect #5 – Benham's disc

The Benham's disc [1, 14] is a circle divided into two equal halves, one white and one black, with the white portion containing several groups of black concentric arcs (see Figure 5). When the disc is rotated the different groups of arcs produce the illusion of color, even under a monochromatic light source. The lengths of the arcs, their relative positions, and the speed at which rotation is performed all contribute to the effect. A complete explanation for this visual effect is lacking, but it has been suggested that it may be related to the growth and decay rates of different color sensations produced by a white stimulus.

For the X11 demonstration, an image of the disc was constructed by drawing a sequence of concentric circular arcs divided into six degree segments, with each segment of each arc drawn with a different color index. The rate at which the color assignment to indices are changed and the lengths and relative positions of the arcs are all under control of sliders with a toggle to reverse the direction of rotation.

The illusion of color can be achieved using our demonstration, but only at relatively high rotation rates (four rotations per second). At these rates, the arcs are changing in four-segment increments between consecutive refresh cycles on a 60 Hz display. This makes the quality of the motion relatively "jumpy" and thus the demonstration is not entirely satisfactory.

Visual effect #6 – Gestalt organization

The final visual effect is based on Gestalt principles [10, 20] and is used to explore the perceptual relationships that exist between independent windows in a display. The demonstration draws two adjacent windows with diagonal lines of identical slope within each window. Within one window, the color of the diagonal line, the background color, the width of the border around the window, and the vertical alignment of the window with respect to the other window are all manipulated by sliders.

Using these parameters, it is possible to adjust the windows so that the two lines are perceived as being either one single line spanning the two windows or two separate lines each contained within its window. The most effective way of breaking the perception of a single line spanning both windows is to increase the vertical misalignment of the lines. Changing the foreground or background colors and changing the horizontal gap between the two windows (while adjusting the vertical gap so that the two lines stay aligned) have a lesser effect in breaking the perception.

Using the Demonstrations as a Benchmark

Although some of the demonstrations were easy to implement using common color lookup table techniques described in the literature [17, 22, 23], it proved surprisingly difficult to achieve satisfactory demonstrations when the visual effects depended not on static versions of the images but on animated images where the display changed with time. The features available in X11 and the display hardware on which it is implemented do not always support the real-time manipulation required.

Some of the lessons painfully learned with the interactive line drawing displays of a decade or two ago have been forgotten in today's raster displays: synchronization of refresh cycles with update cycles, immediate feedback at the lexical and syntactic levels, and adequate refresh rates to match the limits of human visual perception.

Achieving temporal control over animations

Animations (images that change with time) are becoming more important in workstations as the hardware becomes more powerful. The X11 protocol places limitations on the ability of a program to make use of the underlying display hardware. Our demonstrations are a good test of whether X11 provides sufficient support.

Demonstrations such as Benham's disc exist at the limits of practical animation tasks and thus provide a good measure of the performance of X11 and of the workstation displays themselves. The 60-cycle refresh rate is the limiting factor for this effect. Colors are just beginning to appear at the fastest we can rotate the disc. 60 Hz is the limit of the human visual system for temporal information, but to achieve this on a computer display a 120Hz refresh rate is required for animated images to avoid aliasing artifacts. With 60 Hz display, everything above 30 Hz must be filtered to avoid aliasing, whereas with a 120 Hz display filtering only has to be performed above 60 Hz which is higher than the cutoff of the human visual system. Thus a 120 Hz display is sufficient to provide the full range of temporal information that a human can perceive, but a 60 Hz display cannot.

Display processors such as the Adage/Ikonas and the Sun TAAC are capable of refresh rates higher than 60 Hz. We have used them successfully for animated images where the elimination or precise control of visual effects is important [15]. If we believe that animated images will be commonplace in future workstation applications, some provision should be made for higher refresh rates. This is perhaps more important than current efforts to increase the spatial resolution of workstation displays and is a tradeoff that should be explored.

Manipulating the color lookup tables

All of the demonstrations make heavy use of the color lookup table techniques noted earlier because of the relative ease with which animation can be achieved from "static" images when portions of the lookup tables are changed. For the animation to be effective, the lookup tables must be changed at rates consistent with the real-time demands of the human visual system.

We have experimentally determined that the X11 implementation we are using probably updates the lookup table only during the vertical retrace cycle. This is fortunate because it is well known that problems arise with synchronizing updates if this is not done. We were unable to find documentation of this feature for X11. Moreover, we do not know how this is synchronized with the event loop, which means that an X11 client is unlikely to be able to guarantee that lookup table entries are in fact being changed in a precise temporal pattern, such as once every vertical retrace cycle.

The biggest problem is interference from system processes. Our experience is that our lookup table animations have unpredictable performance even on an isolated workstation because of system processes pre-empting the window server. Multiprocessors may be the solution to this, since many workstations now have dedicated processors where the user/client may be able to specify actions that cannot be preempted by the demands of the operating system. But most window systems do not provide a mechanism for specifying this type of behavior. A window server architecture that does support this behavior has been discussed elsewhere [16].

We suggest that all window systems should provide, at least optionally, a method for updating lookup tables in synchronization with the vertical retrace and for delaying further processing of output events until a specified number of vertical retraces have been performed. Early high performance line drawing systems had provisions for this type of synchronization to avoid burning out the CRT as the length of the display list changed. Similar provisions exist in high performance raster displays because double-buffered display algorithms such as the z-buffer [4] have this problem unless the update rate is made oblivious to the complexity of the image.

For Benham's disc and, to a lesser extent the pinwheel, a primary limitation we ran into was the small number of color entries available on our X11 workstations (256 entries). This meant that it was difficult to achieve the appearance of smooth rotation, even within the inner pinwheel. Even with a larger frame buffer such as the 24-bit displays used in high performance workstations, often the lookup table entries are limited to eight bits in each color (the IRIS is a notable exception to this, although it does not permit separate 8-bit lookup tables for the three color guns and thus suffers another limitation).

The vision community has recognized the need for larger lookup tables. A product is now being sold that maps the color output of a 24-bit RGB display to 12-bit monochrome – reversing the original use of lookup tables in 12-bit monochrome systems to provide color – for use in vision experiments.

Localized processing for lexical and syntactic events

A problem related to the uneven performance of lookup table updates that was caused by interference from system processes and a lack of exact synchronization with the refresh cycle is the difficulty of providing simple lexical feedback in X11. Sliders require a significant amount of computation by the client. In X11 it appears to be necessary for a client to "latch on" to the mouse for effective dragging of sliders. This seems inefficient.

NeWS provides a mechanism whereby actions can be performed locally on the server as a result of mouse actions. This same technique existed in the Evans & Sutherland PS 300 through function networks, whereby a dial or mouse could control the position of a visual slider. In many applications, satisfactory animation can be achieved by directly connecting a lookup table entry to a mouse just as for a slider. Older displays featured a hardware cursor, precisely because the turnaround time to communicate with an application program was too long for effective interaction. Similar requirements exist for rapid lexical and syntactic feedback where the necessity of invoking actions within the client is a bottleneck on the real-time performance because the window system is distributed across a network or, in the case of X11, the server protocol does not provide a mechanism for handling important events at higher priority.

Lessons for Windowed Environments

One justification for studying visual effects is the experience gained in the use of windowed environments. The previous section discussed recommendations for future window systems that arise from our experience. This section discusses what we have learned about the proper use of existing windowed environments.

Using visual effects to calibrate displays

Visual effects can provide highly effective techniques for calibrating workstation displays. Calibration is particularly important because many common techniques for improving image quality depend on the calibration of the display system. Antialiasing, dithering, and elimination of Mach bands are examples of display techniques that depend critically on calibration [3, 5]. A proper understanding of the visual effects and the degree to which they can be implemented in a windowed environment can lead to much more effective and efficient calibration techniques.

Nulling of apparent motion seems to be a more sensitive test of equiluminance than other visual techniques [2], though often less easy to implement than direct radiometric techniques [8]. The demonstration we have implemented can be adapted to a calibration technique for achieving equiluminance, which is also required for proper gamma correction. There are two specific applications in which a technique based on nulling of apparent motion would be strongly preferred over radiometry: when a display that is equiluminous for a specific observer is desired; and when radiometry is unavailable or too inconvenient to employ.

Understanding interactions among visual effects

Attempts to control visual effects interact. Reducing one effect may increase another. There is an interaction between spatial and chromatic changes with time. The Benham's disc demonstration illustrates how chromatic changes can be induced by spatial changes in the image (the rotating monochromatic disk is seen to produce color where there is none) and the nulling of apparent motion demonstration illustrates how spatial changes can be induced by chromatic changes (the colored squares provide the illusion of motion where there is none).

The Gestalt organization demonstration illustrates another important class of interactions among visual effects. In attempting to break the illusion of a single diagonal line crossing the boundary between the two windows, the foreground and background colors might be adjusted in the two windows. Yet we know from our study of contrast phenomena that adjusting the foreground affects our perception of the background and that increasing the distance between windows can make it more difficult to detect color differences and hence may decrease the effectiveness of such differences in breaking the illusion of a single line spanning both windows.

Coordinating multiple windows

It is not sufficient to control visual effects with a single window. Multiple windows may interact visually in a variety of ways that can be very unpredictable. Techniques for providing multi-window coordination require an understanding of the visual effects and the mechanisms for controlling them.

Performance in a multi-window environment using X11 cannot be predicted or controlled by individual clients because of the shared single server model. The same is true of the Macintosh. The style used for dragging is a good example of this. Once an object has been "grabbed" by the mouse, the standard technique is to monopolize the cpu (on the Macintosh) while the mouse button is depressed, continually updating the display with the new position of the object. This has the side effect of eliminating any animation the program might be performing at the same time, unless the program is carefully crafted to multiplex the mouse polling with the animation.

For X11, the server model implies that an animated window may suffer degraded performance as a result of other clients interacting or that a client attempting to implement dragging may see sporadic response as a result of other clients performing periodic updates of a window. If the various clients are running on multiple computers connected across a network, it can be very difficult to coordinate their use of the server's limited resources unless the window system protocol supports this directly. Current window systems do not provide this type of capability.

Focusing attention within a display window

The pinwheel demonstration tells us both how to induce the perception of motion in the periphery by making sure that there is a luminance change when displayed objects move and how to inhibit the perception by eliminating any luminance contrast when moving objects are not intended to draw attention to themselves. This means if we want motion to be detected wherever it appears on the screen, there must always be a significant luminance change as a result of the motion, but if we want to restrict the perception of motion to the current focal point, motion should result in only chromatic changes that do not affect the luminance so that the human visual system will ignore motion in the periphery and maintain visual attention at the focal point.

For such a technique to work, the display must be calibrated. Luminance in the periphery is different from luminance in the fovea [7]. This is because the fovea consists only of cones, whereas the periphery involves both rods and cones with a different ratio of red, green, and blue cones from what is present in the fovea. Thus it is important that different luminance values be used if the intention is to balance luminance in the fovea versus the periphery. And it is important that those values be obtained through appropriate calibrations such as those based on visual effects known to occur within the corresponding visual regions. If the red/green balance measured in nulling of apparent motion for squares viewed by the fovea are used as a calibration for suppression of peripheral motion, the results are likely to be unsatisfactory. A more appropriate calibration would be one based on the pinwheel demonstration. This in turn implies that the window system must support the temporal performance necessary to implement that demonstration, even if the end use of the calibration is for an application that does not require that level of performance.

Acknowledgements

The X11 programs were implemented with the assistance of Ian Allen, who provided significant help in understanding the internal workings of Unix/Ultrix and X11. His advice on coding style and portability was invaluable.

The translation for the résumé was provided by Doris Kochanek of the National Film Board of Canada.

This work was partially supported by the Natural Sciences and Engineering Research Council of Canada, the Province of Ontario, and Digital Equipment of Canada Limited.

References

[1] Albers, Josef. *Interaction of Color.* New Haven, CT: Yale University Press, 1963.

[2] Anstis, Stuart M., and Cavanaugh, Patrick. A Minimum Motion Technique for Judging Equiluminance. *Color Vision.* New York, NY: Academic Press, 1983, pp. 155-166.

[3] Barros, J. and Fuchs, H. Generating Smooth 2-D Monocolor Line Drawings on Video Displays. *Computer Graphics* **13**:2 (August, 1979) pp. 260-269.

[4] K. S. Booth, D. R. Forsey and A. W. Paeth, "Hardware assistance for z-buffer visible surface algorithms," *IEEE Computer Graphics and Applications* **6**:11 (November, 1986) pp. 31-39.

[5] Catmull, E. A Tutorial on Compensation Tables. *Computer Graphics* **13**:2 (August, 1979) pp. 1-7.

[6] Cavanaugh, Patrick, personal communication.

[7] Cavanaugh, Patrick, MacLeod Donald I. A., and Antis, Stuart M. Equiluminance: Spatial and Temporal Factors and the Contribution of Blue-sensitive Cones. *Journal of the Optical Society of America A* **4**:8, pp. 1428-1438, August 1987.

[8] Cowan, W. B., Colorimetric Properties of Video Displays. Course Notes from the *Annual Meeting of the Optical Society of America*, Orlando, Florida, October, 1989.

[9] Goetz, S.M. and Beatty, J.C. Colour Principles and Experience for Computer Graphics. *Graphics Interface '82*, (May, 1982) pp. 313-322.

[10] Hochberg, Julian E. *Perception.* Englewood Cliffs, NJ: Prentice-Hall, 1978.

[11] Jones, Oliver. *Introduction to the X Window System.* Englewood Cliffs, NJ: Prentice-Hall, 1989.

[12] Kaehler, C., Horn, B., Capps, S. *Macintosh.* Cupertino, CA: Apple Computer Inc., 1984.

[13] Kaiser, Peter A., Vimal, Ram L. P., Cowan, William B., and Hibino, Haruo. Nulling of Apparent Motion as a Method for Assessing Sensation Luminance: An Additivity test. *Color Research and Application,* **14**:4, pp. 187-191, August 1989.

[14] Kaufman, Lloyd. *Sight and Mind.* New York, NY: Oxford University Press, 1974. pp. 137-141.

[15] R. V. Klassen, PhD thesis, University of Waterloo, 1989.

[16] J. V. Kelley, K. S. Booth and M. Wein, "Design experience with a multiprocessor window system architecture," *Graphics Interface '89* (June, 1989) pp. 62-69.

[17] S. A. MacKay and K. S. Booth, "Techniques for frame buffer animation," *Graphics Interface '82*, (May, 1982) pp. 213-220.

[18] Luckiesh, M. *Visual Illusions: Their Causes, Characteristics and Applications.* New York, NY: Dover Publications, 1965.

[19] Massachusetts Institute of Technology. *X Window System Protocol, Version 11.* September 1987.

[20] Rock, Irvin. *An Introduction to Perception.* New York, NY: Macmillan Publishing, 1975.

[21] Scheifler, Robert W., Gettys, James, and Newman, Ron. *X Window System.* Bedford, MA: Digital Press, 1988.

[22] Shoup, Richard G. Color Table Animation. *Computer Graphics (SIGGRAPH '79 Conference Proceedings.)* **13**:2, pp. 8-13, August 1979.

[23] Sloan, K. R. Jr., and Brown, C. M. Color Map Techniques *Computer Graphics and Image Processing* pp. 297-317, August 1979.

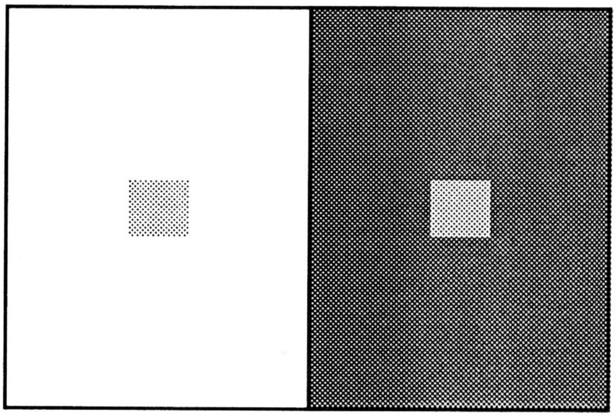

Figure 1. Black/white contrast.

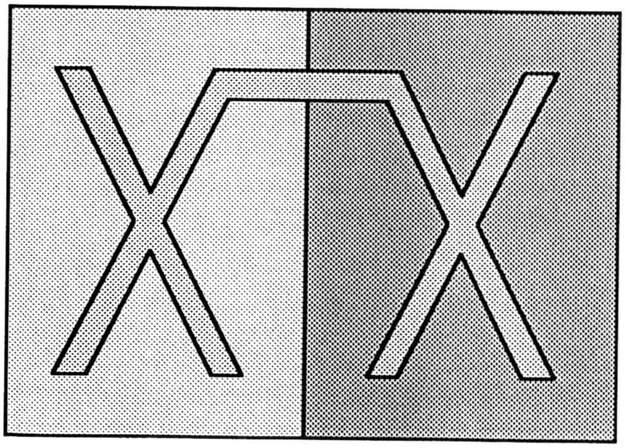

Figure 2. Yellow/gray contrast.

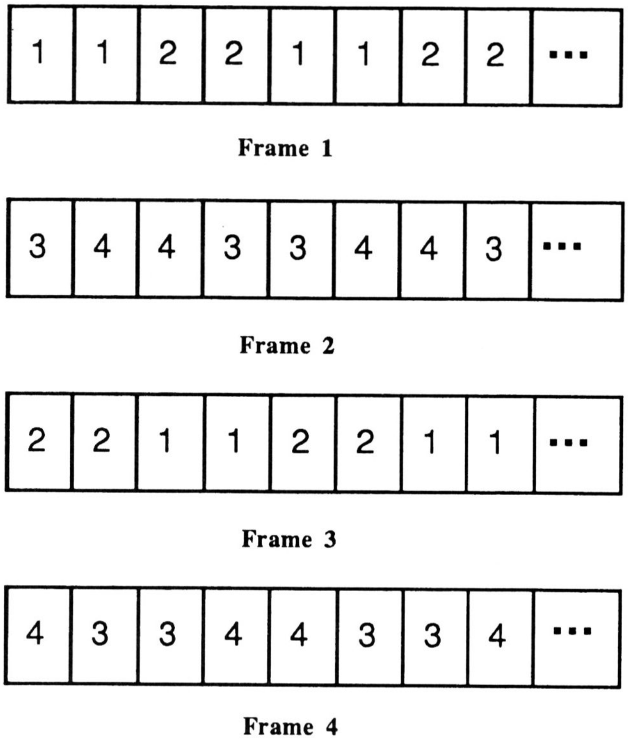

Figure 3. Apparent motion from color cycling.

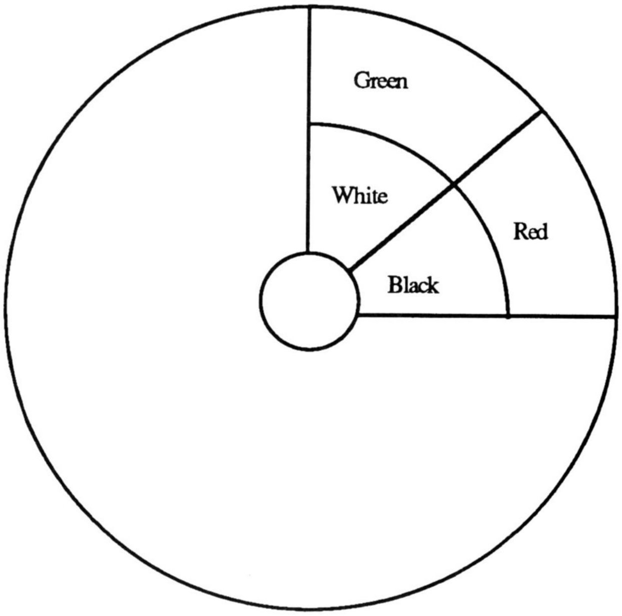

Figure 4. The peripheral pinwheel.

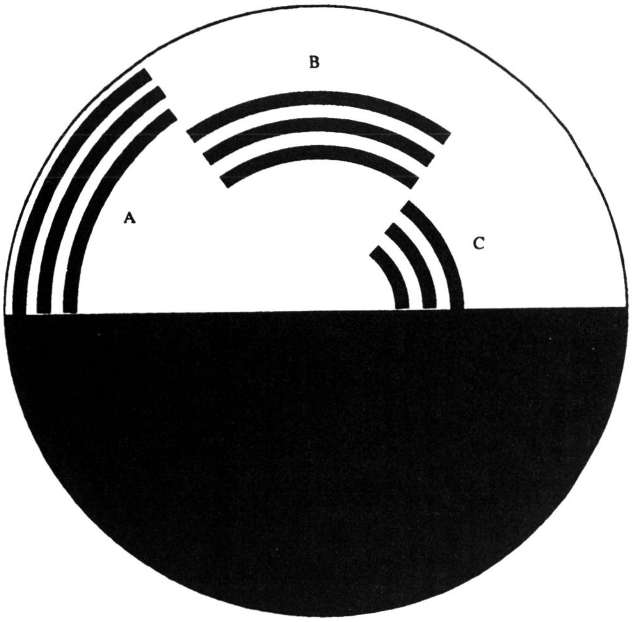

Figure 5. Benham's disc.

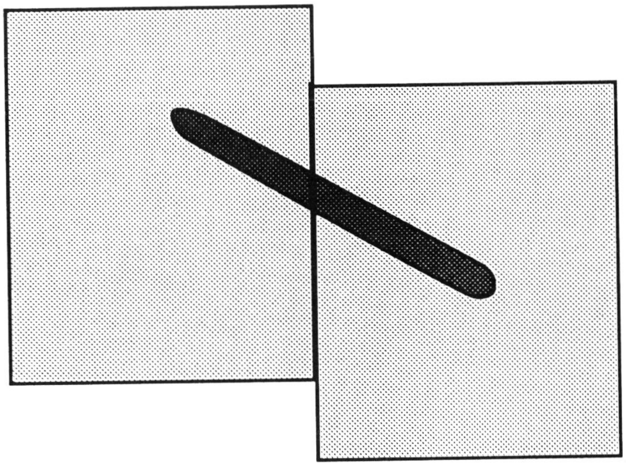

Figure 6. Gestalt organization (interwindow interference).