

Our Peripheral Vision is Poor

Chapter 4 explained how the human visual system differs from a digital camera in the way it detects and processes color. Our visual system also differs from a camera in its resolution. On a digital camera's photosensor, photoreceptive elements are spread uniformly in a tight matrix, so spatial resolution is constant across the entire image frame. The human visual system is not like that.

This chapter explains why:

- Stationary items in muted colors presented in the periphery of a person's visual field often will not be noticed.
- Motion in the periphery is usually noticed.

RESOLUTION OF THE FOVEA COMPARED WITH THE PERIPHERY

The spatial resolution of the human visual field drops greatly from the center to the edges for three reasons:

- **Pixel density.** Each eye has six to seven million retinal cone cells. They are packed much more tightly in the center of our visual field—a small region called the *fovea*—than at the edges of the retina (see Fig. 5.1). The fovea has about 158,000 cone cells in each square millimeter. The rest of the retina has only 9000 cone cells per square millimeter.
- **Data compression.** Cone cells in the fovea connect 1:1 to the ganglial neuron cells that begin the processing and transmission of visual data, while elsewhere on the retina, multiple photoreceptor cells (cones and rods) connect to each ganglion cell. In technical terms, information from the visual periphery is compressed (with data loss) before transmission to the brain, while information from the fovea is not compressed.

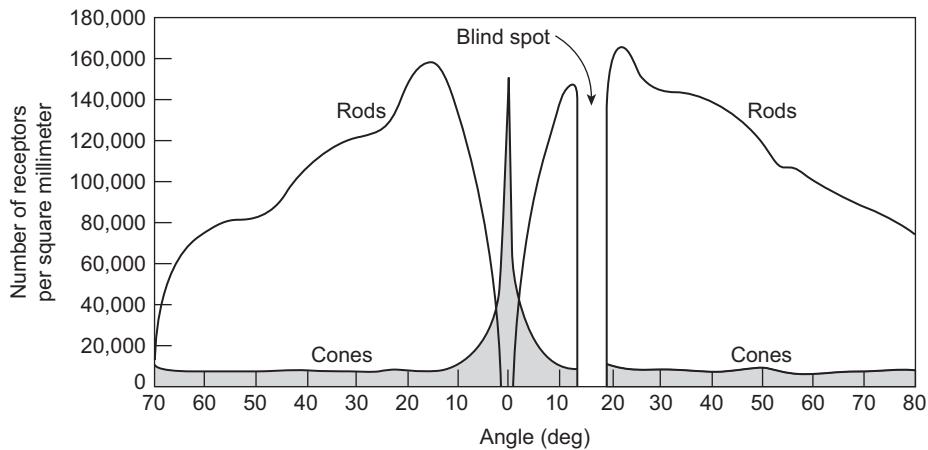


FIGURE 5.1

Distribution of photoreceptor cells (cones and rods) across the retina. (From Lindsay, P., Norman, D.A., 1972. *Human Information Processing*. Academic Press, New York and London.)

- Processing resources.** The fovea is only about 1% of the retina, but the brain's visual cortex devotes about 50% of its area to processing input from the fovea. The other half of the visual cortex processes data from the remaining 99% of the retina.

The result is that our vision has much, much greater resolution in the center of our visual field than elsewhere (Lindsay and Norman, 1972). Stated in developer jargon: in the center 1% of your visual field (i.e., the fovea), you have a high-resolution TIFF, and everywhere else, you have only a low-resolution JPEG. That is *nothing like* a digital camera.

To visualize how small the fovea is compared with your entire visual field, hold your arm straight out and look at your thumb. Your thumbnail, viewed at arm's length, corresponds approximately to the fovea (Ware, 2008). While you have your eyes focused on the thumbnail, everything else in your visual field falls outside your fovea.

In the fovea, people with normal vision have very high resolution: they can resolve several thousand dots within that region—better resolution than many of today's pocket digital cameras. Just outside the fovea, the resolution is already down to a few dozen dots per inch viewed at arm's length. At the edges of our vision, the “pixels” of our visual system are as large as a melon (or human head) at arm's length (see Fig. 5.2).

Even though our eyes have more rods than cones—125 million versus 6 to 7 million—peripheral vision has much lower resolution than foveal vision. This is because while most of our cone cells are densely packed in the fovea (in 1% of the retina's area), the rods are spread out over the rest of the retina (in 99% of the retina's area). In people with normal vision, peripheral vision is about 20/200, which in the United States is considered legally blind. Think about that: in the periphery of your visual

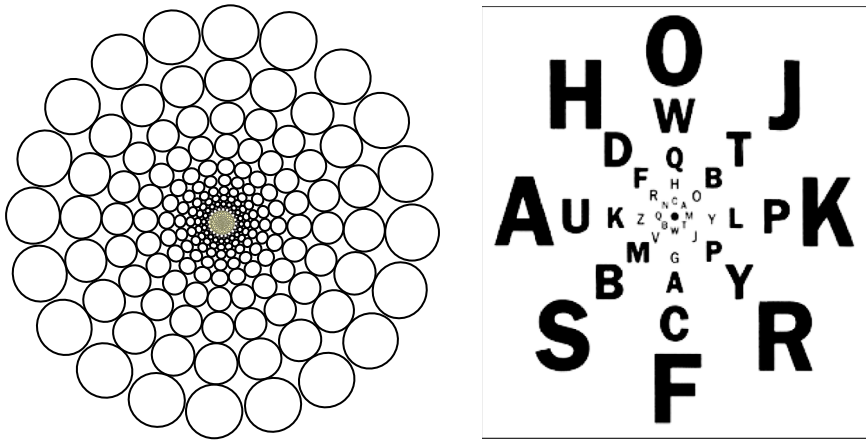


FIGURE 5.2

The resolution of our visual field is high in the center but much lower at the edges. (Right image from Vision Research, vol. 14, 1974. Elsevier.)

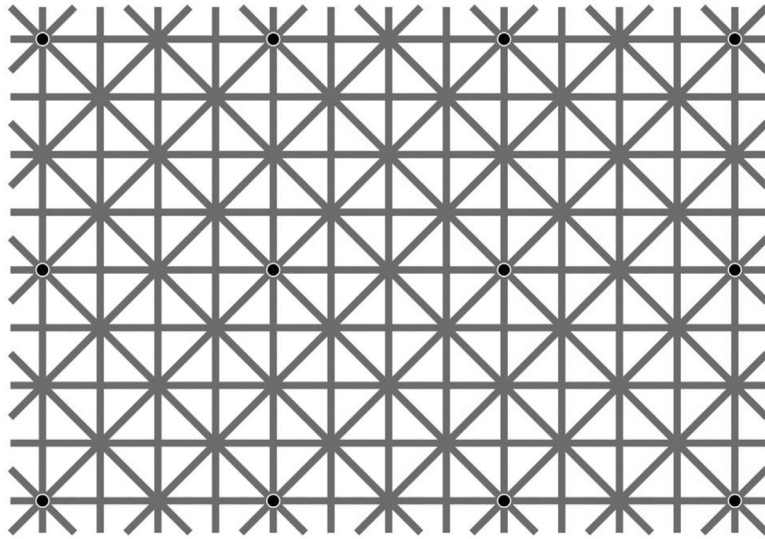
field—which is most of it—you are legally blind. Here is how brain researcher David Eagleman (2012, p. 23) describes it:

The resolution in your peripheral vision is roughly equivalent to looking through a frosted shower door, and yet you enjoy the illusion of seeing the periphery clearly.... Wherever you cast your eyes appears to be in sharp focus, and therefore you assume the whole visual world is in focus.

If our peripheral vision has such low resolution, one might wonder why we don't see the world in a kind of tunnel vision where everything is out of focus except what we are directly looking at now. Instead, we seem to see our surroundings sharply and clearly all around us. We experience this illusion because our eyes move rapidly and constantly about three times per second even when we don't realize it, focusing our fovea on selected pieces of our environment. Our brain fills in the rest in a gross, impressionistic way based on what we know and expect.¹ Our brain does not have to maintain a high-resolution mental model of our environment because it can order the eyes to sample and resample details in the environment as needed (Clark, 1998).

For example, as you read this page, your eyes dart around, scanning and reading. No matter where on the page your eyes are focused, you have the impression of viewing a complete page of text because, of course, you are.

¹Our brains also fill in perceptual gaps that occur during rapid (saccadic) eye movements, when vision is suppressed (see Chapter 14).

**FIGURE 5.3**

Ninio's extinction illusion. You see dots only by looking directly at them.

But now imagine this: you are viewing the page on a computer screen. The computer is tracking your eyes and knows where your fovea is focused. The computer shows the correct text for that spot, but everywhere else the computer shows random meaningless text. As your fovea zips around the page, the computer instantly updates each area where your fovea pauses to show the correct text there, while the previous position of your fovea returns to textual noise. Amazingly, experiments have shown that people *rarely notice* this: not only can they read, they believe that they are viewing a full page of meaningful text (Clark, 1998). However, it does slow people's reading even if they don't realize it (Larson, 2004).

Perhaps the best demonstration that our peripheral vision does not see as much detail as our foveal vision is Ninio's Extinction Illusion (see Fig. 5.3). Try to count the dots. Only dots that are in the center of your visual field are visible even though the entire grid appears to be in sharp focus. This shows that what we perceive as high-resolution peripheral vision is actually an artificial construction of our visual system filling in what it expects to be there.

The fact that retinal cone cells are distributed tightly in and near the fovea, and sparsely in the periphery of the retina, affects not only spatial resolution but also color resolution. We can discriminate colors better in the center of our visual field than at the edges.

Another interesting fact about our visual field is that it has a gap—a small area (blind spot) in which we see nothing. The gap corresponds to the spot on our retina where the optic nerve and blood vessels exit the back of the eye (see Fig. 5.1). There are no retinal rod or cone cells at that spot, so when the image of an object in our

**FIGURE 5.4**

To “see” the retinal gap, cover your left eye, hold this book near your face, and focus your right eye on the +. Move the book slowly away from you, staying focused on the +. The @ will disappear at some point.

visual field happens to fall on that part of the retina, we don’t see it. We usually don’t notice this hole in our vision because our brain fills it in with the surrounding content, like a graphic artist using Photoshop to fill in a blemish on a photograph by copying nearby background pixels.

People sometimes experience the blind spot when they gaze at stars. As you look at one star, a nearby star may disappear briefly into the blind spot until you shift your gaze. You can also observe the gap by trying the exercise in [Fig. 5.4](#). Some people have other gaps resulting from imperfections on the retina, retinal damage, or brain strokes that affect the visual cortex,² but the optic nerve gap is an imperfection everyone shares.

IS THE VISUAL PERIPHERY GOOD FOR ANYTHING?

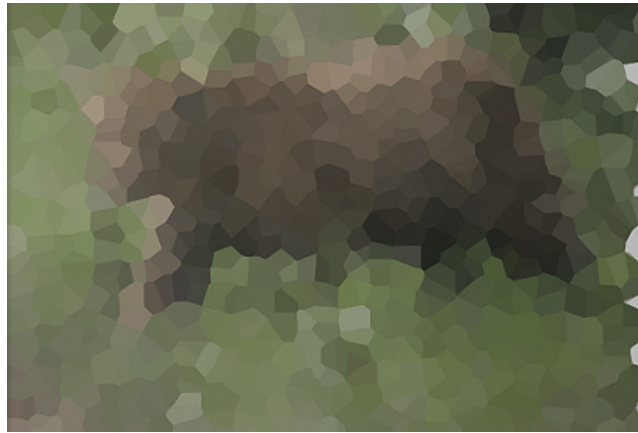
It seems that the fovea is better than the periphery at just about everything. One might wonder why we have peripheral vision. What is it good for? Our peripheral vision serves three important functions: it guides our fovea toward objects and events that match our goals, it detects motion and guides our fovea there, and it lets us see better in the dark.

Function 1: guides fovea toward objects matching goals

Remember what [Chapter 1](#) explained: our perception is biased by our goals. Peripheral vision provides low-resolution cues to guide our eye movements so that our fovea visits the interesting and crucial parts of our visual field. Our eyes don’t scan our environment randomly. They move so as to focus our fovea on important things—things related to our goals and to possible threats—the most important ones (usually) first. Think of peripheral vision as our visual system’s “forward patrol,” primed to notice important things and report them to “central command.” Thus, the fuzzy cues in the periphery of our visual field provide the data that helps our brain plan where to move our eyes and attention.

For example, when we scan a medicine label for a “use by” date, a fuzzy blob in the periphery with the vague form of a date is enough to cause an eye movement that lands the fovea there to allow us to check it. If we are browsing a produce market

²See VisualSimulations.com.

**FIGURE 5.5**

A moving shape at the edge of our vision draws our eye: it could be food, or it might consider us food.

looking for strawberries, a blurry reddish patch at the edge of our visual field draws our eyes and our attention, even though sometimes it will be radishes instead of strawberries. If we hear an animal growl nearby, a fuzzy animal-like shape in the corner of our eye will be enough to zip our eyes in that direction, especially if the shape is moving toward us (see [Fig. 5.5](#)). And as shown in [Chapter 1](#), where we look on a web page or app screen depends on what on the display matches our goals.

How peripheral vision guides and augments central, foveal vision is discussed more in the “[Visual Search is Linear Unless Targets ‘Pop’ in the Periphery](#)” section later in this chapter.

Function 2: detects motion

A related guiding function of peripheral vision is that it is good at detecting motion. Anything that moves in our visual periphery, even slightly, is likely to draw our attention—and hence our fovea—toward it. The reason for this phenomenon is that our ancestors—including prehuman ones—were selected for their ability to spot food and avoid predators. As a result, even though we can move our eyes under conscious, intentional control, some of the mechanisms that control where our eyes look are preconscious, involuntary, and very fast.

What if we have no reason to expect that there might be anything interesting in a certain spot in the periphery,³ and nothing in that spot attracts our attention? Our eyes may never move our fovea to that spot, so we may never see what is there.

³ See Chapter 1 on how expectations bias our perception.

Function 3: lets us see better in the dark

A third function of peripheral vision is to allow us to see in low-light conditions—for example, on starlit nights, in caves, around campfires, etc. These were conditions under which vision evolved and in which people—like the animals that preceded them on Earth—spent much of their time until the invention of the electric light bulb in the 1800s.

Just as the rods are less functional in well-lighted conditions (see [Chapter 4](#)), the cones don't function very well in low light, so our rods take over. Low-light, rods-only vision is called *scotopic vision*. An interesting fact is that because there are no rods in the fovea, you can see objects better in low-light conditions (e.g., faint stars) if you don't look directly at them.

EXAMPLES FROM COMPUTER USER INTERFACES

The low acuity of our peripheral vision explains why users of software and websites fail to notice error messages in some applications and websites. When someone clicks a button or link, that is usually where his or her fovea is positioned. Everything on the screen not within 1–2 cm of the click location (assuming a normal computer viewing distance) is in peripheral vision, where resolution is low. If after the click, an error message appears in the periphery, it should not be surprising if the person does not notice it.

For example, at Informaworld.com, the former online publication website of Informa Healthcare, if a user entered an incorrect username or password and clicked “Sign In,” an error message appeared in a “message bar” far away from where the users’ eyes were most likely focused (see [Fig. 5.6](#)). The red word “Error” might appear in the user’s peripheral vision as a small reddish blob, which would help draw the eyes in that direction. However, the red blob might fall into a gap in the viewer’s visual field and not be noticed at all.

An alert message displayed by Delta Airlines’ (2017) website is similarly easy to miss (see [Fig. 5.7](#)). Customers sometimes encountered it when clicking through seat-selection steps of the flight check-in process. Users’ attention almost certainly would be focused on the Previous Flight/First Flight buttons at the lower right of the screen. The alert message is not as far from where site visitors would be looking as for the one at Informaworld.com, but it is tiny and insufficiently highlighted.

Consider the sequence of events from a user’s point of view. The user enters a username and password and then clicks “Sign In.” The page redisplay with blank fields. The user thinks “Huh? I gave it my login information and hit ‘Sign In,’ didn’t I? Did I hit the wrong button?” The user reenters the username and password and clicks “Sign In” again. The page redisplay with empty fields again. Now the user is really confused. The user sighs (or curses), sits back in his chair, and lets his eyes scan the screen. Suddenly noticing the error message, the user says “A-ha! Has that error message been there all along?”



FIGURE 5.6

The error message from the former Taylor & Francis Informaworld website (2010) for a faulty login appeared in peripheral vision, where most users would probably not see it.

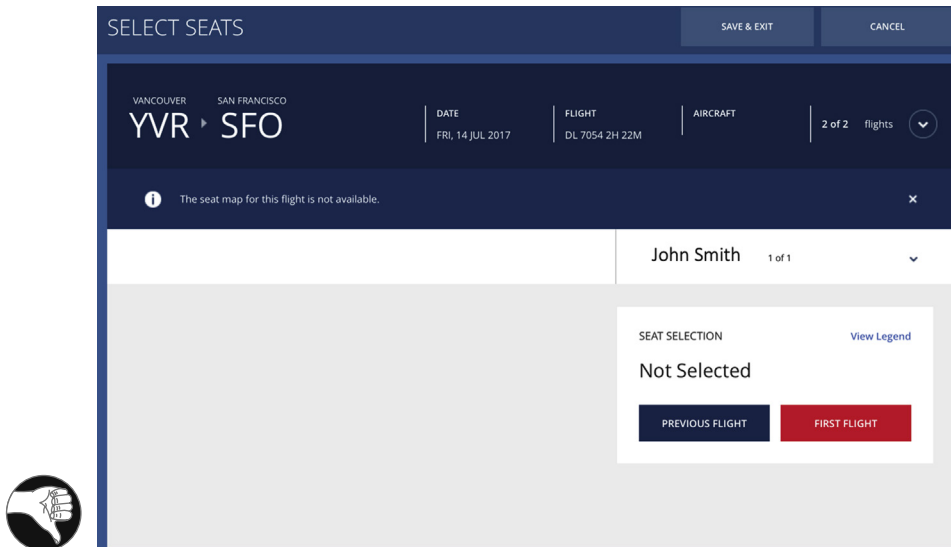
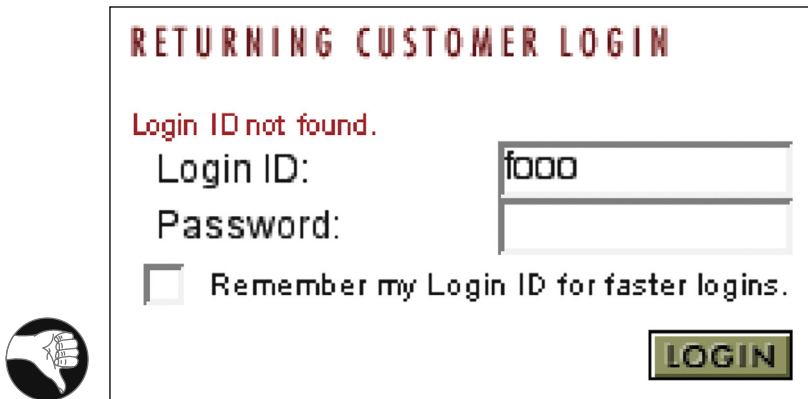


FIGURE 5.7

Error message at Delta.com (2017). Can you spot it?

**FIGURE 5.8**

This error message for an invalid login was easy to miss even though it is not far from the “Login” button. Why?

**FIGURE 5.9**

Simulation of a user's visual field while the fovea is fixed on the “Login” button.

Even when an error message is placed nearer to the center of the viewer's visual field than in the preceding examples, other factors can diminish its visibility. For example, from about 2003 to 2008,⁴ the website of Airborne Express (now part of DHL) signaled a login failure by displaying an error message in red just above the Login ID field (see Fig. 5.8). The error message was entirely in red and fairly near the “Login” button where a user's eyes and attention would be focused. Can you think of any reasons people might not initially see it?

⁴Although this example is old, it is still the best example of this issue I have found.





One reason is that the error message was still in peripheral vision, not the fovea. The fovea is small: just a centimeter or two on a computer screen, assuming the user is a normal distance from the screen.

A second reason is that the error message was not the only red thing near the top of the page. The page title was also red. Resolution in the periphery is very low. When the error message appeared, a user's peripheral vision might not register any change—there was a red blotch there before, and with the error message, there still was (see Fig. 5.9).

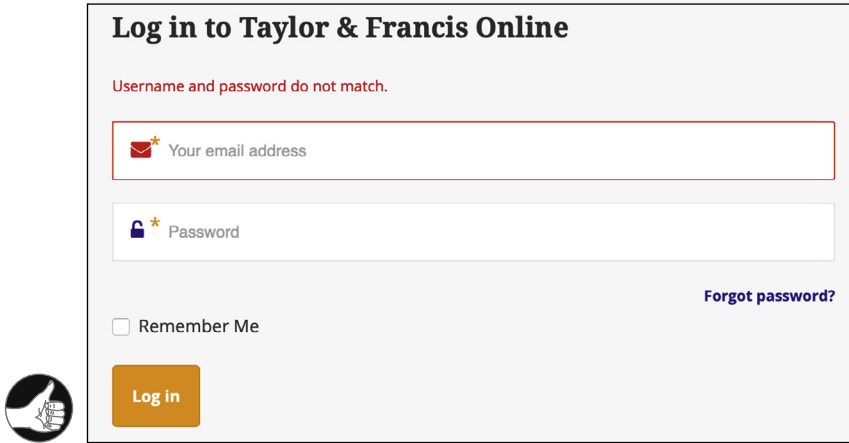
If the page title were black or any other color besides red, the red error message would be more likely to be noticed, even though it appeared in the periphery of the user's visual field.

COMMON METHODS OF MAKING MESSAGES VISIBLE

Several common and well-known methods can ensure that error messages are seen:


- **Put it where users are looking.** People focus on predictable places when interacting with graphical user interfaces. In Western societies, people tend to traverse forms and control panels from upper left to lower right. While moving the screen pointer, people usually look either at where it is or where they are moving it to. When people click a button or link, they can usually be assumed to be looking directly at it, at least for a few moments afterward. Designers can use this predictability to position error messages near where they expect users to be looking.
- **Mark the error.** Somehow, mark the error prominently to indicate clearly that something is wrong. Often this can be done by simply placing the error message near what it refers to, unless that would place the message too far from where users are likely to be looking.
- **Use an error symbol.** Make errors or error messages more visible by marking them with an error symbol, such as , , , or .
- **Reserve red for errors.** By convention, interactive computer systems use the color red to connote *alert*, *danger*, *error*, etc. Using red for any other information on a computer display risks misinterpretation. But suppose you are designing a website for Stanford University, which has red as its school color. Or suppose you are designing for a Chinese market, where red is considered an auspicious, positive color. What do you do? Use another color for errors, mark them with error symbols, or use stronger methods (see the next section).


The most recent Taylor & Francis login error screen uses several of these techniques (see Fig. 5.10). Compare it with their earlier website (see Fig. 5.6). A bigger, bolder font for the error message and an error symbol would improve this, but it is much better than in the 2010 site.



Log in to Taylor & Francis Online

Username and password do not match.

 Your email address

 Password

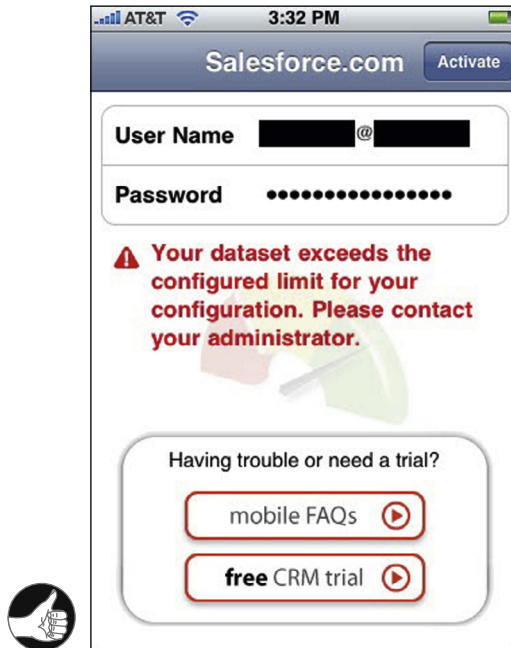
☐ Remember Me

[Forgot password?](#)

Log in

FIGURE 5.10

In the latest Taylor & Francis website (2020), the error message for a faulty login is displayed near where users will be looking, and both the error message and the login field are highlighted in red (compare with Fig. 5.6).



AT&T 3:32 PM


Salesforce.com [Activate](#)

User Name [redacted]@[redacted]

Password [redacted]

⚠ Your dataset exceeds the configured limit for your configuration. Please contact your administrator.

Having trouble or need a trial?

[mobile FAQs](#) 


[free CRM trial](#) 

FIGURE 5.11

Salesforce.com's mobile site displays error messages prominently, midscreen.

Salesforce.com's mobile app displays error messages in a way that makes them hard to miss (see Fig. 5.11). They are marked with an error symbol and displayed in red in the middle of the screen.

HEAVY ARTILLERY FOR MAKING USERS NOTICE MESSAGES

If the common, conventional methods of making users notice messages are not enough, three stronger methods are available to user-interface designers: a pop-up message in an error dialogue box, use sound (e.g., a beep), and wiggle or blink briefly. However, these methods, while very effective, have significant negative effects, so they should be used sparingly and with great care.

Method 1: a pop-up message in an error dialogue box

Displaying an error message in a dialogue box sticks it right in the user's face, making it hard to miss. Error dialogue boxes interrupt the user's work and demand immediate attention. That is good if the error message signals a critical condition, but it can annoy people if such an approach is used for a minor message, such as confirming the execution of a user-requested action.

The annoyance of pop-up messages rises with the degree of modality. *Nonmodal* pop-ups allow users to ignore them and continue working. *Application-modal* pop-ups block any further work in the application that displayed the error but allow users to interact with other software on their computer. *System-modal* pop-ups block any user action until the dialogue has been dismissed.

Application-modal pop-ups should be used sparingly—for example, only when application data may be lost if the user doesn't attend to the error. System-modal pop-ups should be used extremely rarely—basically only when the system is about to crash, taking hours of work with it, or if people will die if the user misses the error message.

On the Web, an additional reason to avoid pop-up error dialogue boxes is that some people set their browsers to block *all* pop-up windows. If your website relies on pop-up error messages, some users may never see them.

REI.com has an example of a pop-up dialogue being used to display an error message. The message is displayed when someone who is registering as a new customer omits any required field in the form (see Fig. 5.12). Is this an appropriate use of a pop-up dialogue? The improved Taylor & Francis error message display (see Fig. 5.10) shows that data entry errors can be signaled well without pop-up dialogues, so REI.com's use of them seems a bit heavy-handed.

Examples of more appropriate use of error dialogue boxes come from Microsoft Excel (see Fig. 5.13A) and Adobe InDesign (see Fig. 5.13B). In both cases, loss of data is at stake.

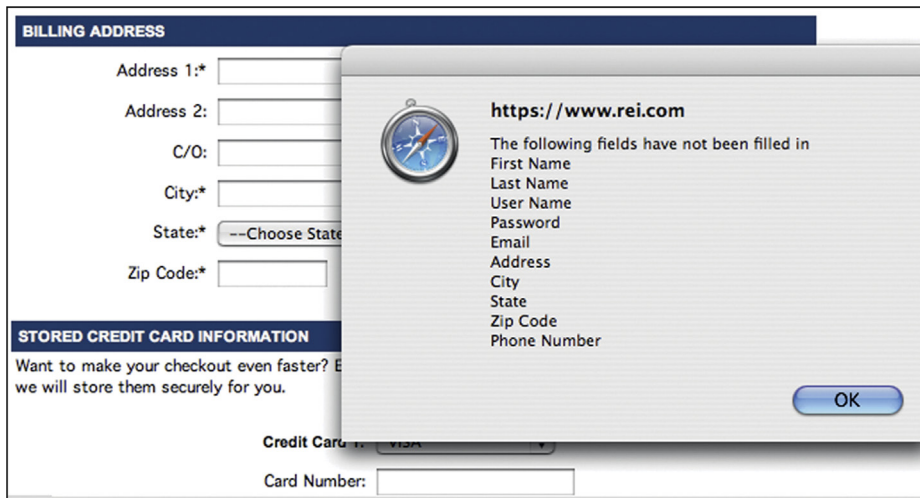


FIGURE 5.12

REI.com's pop-up dialogue box signals required data that was omitted. It is hard to miss, but perhaps overkill.

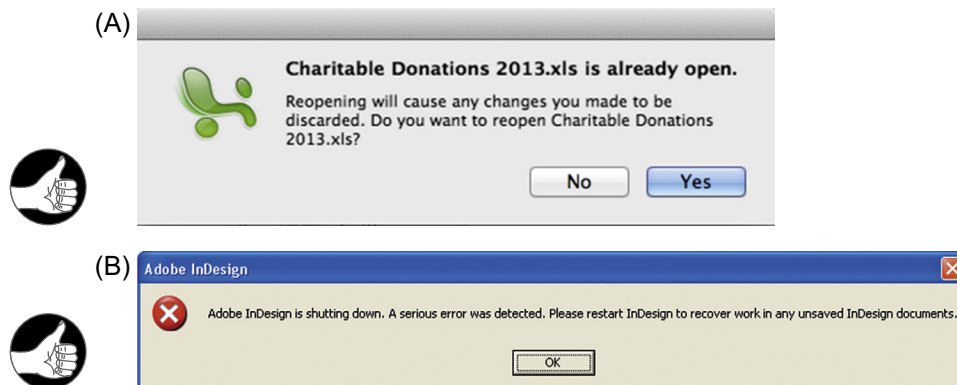


FIGURE 5.13

Appropriate pop-up error dialogues: (A) Microsoft Excel and (B) Adobe InDesign.

Method 2: use sound (e.g., a beep)

When a computer beeps, it tells its user something has happened that requires attention. The person's eyes reflexively begin scanning the screen for whatever caused the beep. This can allow users to notice an error message someplace other than where they were just looking, such as in a standard error message box on the display. That is the value of beeping.

However, imagine many people in a cubicle work environment or a classroom, all using an application that signals all errors and warnings by beeping. Such a workplace would be very annoying to say the least. Worse, people would not be able to tell whether their own computer or someone else's was beeping.

The opposite situation is noisy work environments (e.g., factories or computer server rooms), where auditory signals emitted by an application might be masked by ambient noise. Even in nonnoisy environments, some computer users simply prefer quiet and mute the sound on their computers or turn it way down.

For these reasons, signaling errors and other conditions with sound are remedies that can be used only in very special, controlled situations.

Computer games often use sound to signal events and conditions. In games, sound isn't annoying; it is expected. Its use in games is widespread, even in game arcades where dozens of machines are all banging, roaring, buzzing, clanging, beeping, and playing music at once. (Well, it is annoying to parents who have to go into the arcades and endure all the screeching and booming to retrieve their kids, but the games aren't designed for them.)

Method 3: wiggle or blink briefly

As described earlier in this chapter, our peripheral vision is good at detecting motion, and motion in the periphery causes reflexive eye movements that bring the motion into the fovea. User-interface designers can make use of this by wiggling or flashing messages briefly when they want to ensure that users see them. It does not take much motion to trigger eye movement toward it. Just a tiny bit of motion is enough to make a viewer's eyes zip over in that direction. Millions of years of evolution have had quite an effect.

As an example of using motion to attract users' eye attention, Apple's iCloud online service briefly shakes the entire dialogue box horizontally when a user enters an invalid username or password (see [Fig. 5.14](#)). In addition to clearly indicating "No" (like a person shaking his head), this attracts the users' eyeballs, guaranteed. Because after all, the motion in the corner of your eye might be a leopard.

The most common use of blinking in computer user interfaces (other than advertisements) is in menu bars. When an action (e.g., Edit or Copy) is selected from a menu, it usually blinks once before the menu closes to confirm that the system "got" the command—that is, that the user didn't miss the menu item. This use of blinking is very common. It is so quick that most computer users aren't even aware of it, but if menu items didn't blink once, we would have less confidence that we actually selected them.

Motion and blinking, like pop-up dialogue boxes and beeping, must be used sparingly. Most experienced computer users consider wiggling, blinking objects on-screen to be annoying. Most of us have learned to ignore displays that blink because many such displays are advertisements. Conversely, a few computer users have attentional impairments that make it difficult for them to ignore something blinking or wiggling.



FIGURE 5.14

Apple's iCloud shakes the dialogue box briefly on login errors to attract a user's fovea toward it.

Therefore, if wiggling or blinking is used, it should be brief—it should last about a quarter- to a half-second, no longer. Otherwise, it quickly goes from an unconscious attention-grabber to a conscious annoyance.

Use heavy-artillery methods sparingly to avoid habituating your users

There is one final reason to use the preceding heavy-artillery methods sparingly (i.e., only for critical messages): to avoid *habituating* your users. When pop-ups, sound, motion, and blinking are used too often to attract users' attention, a psychological phenomenon called *habituation* sets in (see [Chapter 1](#)). Our brain pays less and less attention to any stimulus that occurs frequently.

It is like the old fable of the boy who cried “Wolf!” too often. Eventually, the villagers learned to ignore his cries, so when a wolf actually did come, his cries went unheeded. Overuse of strong attention-getting methods can cause important messages to be blocked by habituation.

VISUAL SEARCH IS LINEAR UNLESS TARGETS “POP” IN THE PERIPHERY

As explained earlier, one function of peripheral vision is to drive our eyes to focus the fovea on important things—those that match our goals or that might be a threat. For example, objects moving in our peripheral vision fairly reliably “pull” our eyes and

```

L Q R B T J P L F B M R W S
F R N Q S P D C H K U T
G T H U J L U 9 J V Y I A
E X C F T Y N H T D O L L 8
G V N G R Y J G Z S T 6 S
3 L C T V B H U S E M U K
W Q E L F G H U Y I K D 9

```

FIGURE 5.15

Finding the Z requires scanning carefully through the characters.

attention toward them. Similarly, fuzzy, blurred images related to our current goals also attract our attention. This is why our perception—in this case our visual perception—is biased by our goals (see [Chapter 1](#)).

Peripheral vision is a *crucial* component in visual search despite its low spatial and color resolution. When we are looking for an object, our entire visual system, including the periphery, primes itself to detect that object. It does so by sensitizing neural networks running from the retina to the visual cortex of the brain. The neural networks are sensitized to detect *features* of the sought object (Treisman and Gelade, 1980; Wolfe, 1994; Wolfe and Gray, 2007).

However, how helpful the periphery is in aiding visual search depends strongly on the identifying features of the sought object and how distinct those features are from the features of other objects in our visual field. Look quickly at [Fig. 5.15](#) and find the Z.

To find the Z, you had to scan carefully through the characters until your fovea landed on it. In the lingo of vision researchers, the time to find the Z is *linear*: it depends approximately linearly on the number of distracting characters and the position of the Z among them. Why? The features that define the shape of the letter Z—vertical lines and a diagonal line—do not distinguish it from the surrounding letters, so our peripheral vision cannot spot it. We can only find it when our fovea lands on it. In designer jargon, letter shape does not “pop out” (“pop” for short) in peripheral vision.

Now look quickly at [Fig. 5.16](#) and find the bold character.

That was much easier (i.e., faster), wasn’t it? You did not have to scan your fovea carefully through all the characters. Your periphery quickly detected the boldness and determined its location, and because that is what you were seeking, your visual system moved your fovea there. Your periphery could not determine exactly *what* was bold—that is beyond its resolution and abilities—but it did locate the boldness. In vision-researcher lingo, the periphery was primed to look for boldness in parallel over its entire area, and boldness is a distinctive feature of the target, so searching for a bold target is nonlinear. In designer lingo, we say that boldness “pops” in the periphery, assuming that only the target is bold.

Color “pops” even more strongly. Compare counting the L’s in [Fig. 5.17](#) with counting the red characters in [Fig. 5.18](#).

G T H U J L U 9 J V Y I A
 L Q R B T J P L F B M R W S
 3 L C T V B H U S E M U K
 F R N Q S P D C H K U T
 W Q E L F G H B Y I K D 9
 G V N **G** R Y J G Z S T 6 S
 E X C F T Y N H T D O L L 8

FIGURE 5.16

Finding the bold letter does not require scanning through everything.

L Q R B T J P L F B M R W S
 F R N Q S P D C H K U T
 G T H U J L U 9 J V Y I A
 E X C F T Y N H T D O L L 8
 3 L C T V B H U S E M U K
 G V N G R Y J G Z S T 6 S
 W Q E L F G H U Y I K D 9

FIGURE 5.17

Counting L's is hard—letter shape doesn't “pop” among characters.

W Q E L F G H U Y I K D 9
 F R N Q S P D C H K U T
 3 L C T V B H U S E M U K
 G T H U J L U 9 J V Y I A
 L Q R B T J P L F B M R W S
 E X C F T Y N H T D O L L 8
 G V N G R Y J G Z S T 6 S

FIGURE 5.18

Counting red characters is easy because color “pops.”

What else makes things “pop” in the periphery? As described earlier, the periphery easily detects motion, so motion “pops.” Generalizing from *boldness*, we also can say that font *weight* “pops,” because if all but one of the characters on a display were bold, the *nonbold* character would stand out. In general, a visual target will “pop out” in your periphery if it differs from surrounding objects in features that the low-resolution peripheral vision can detect. The more distinctive the features of the target, the more it “pops,” assuming the periphery can detect those features.

Using peripheral “pop” in design

Designers use peripheral “pop” to focus the attention of a product’s users as well as to allow users to find information faster. [Chapter 3](#) described how visual hierarchy—titles, headings, boldness, bullets, and indenting—can make it easier for users to spot and extract the information they need from text. Glance back at Fig. 3.7B in [Chapter 3](#) and see how the headings and bullets make the topics and subtopics “pop” so readers can go right to them.

Many interactive systems use color to indicate status, usually reserving red for problems. Online maps and most vehicle GPS devices mark traffic jams with red so they stand out (see [Fig. 5.19](#)). Systems for controlling air traffic mark potential collisions

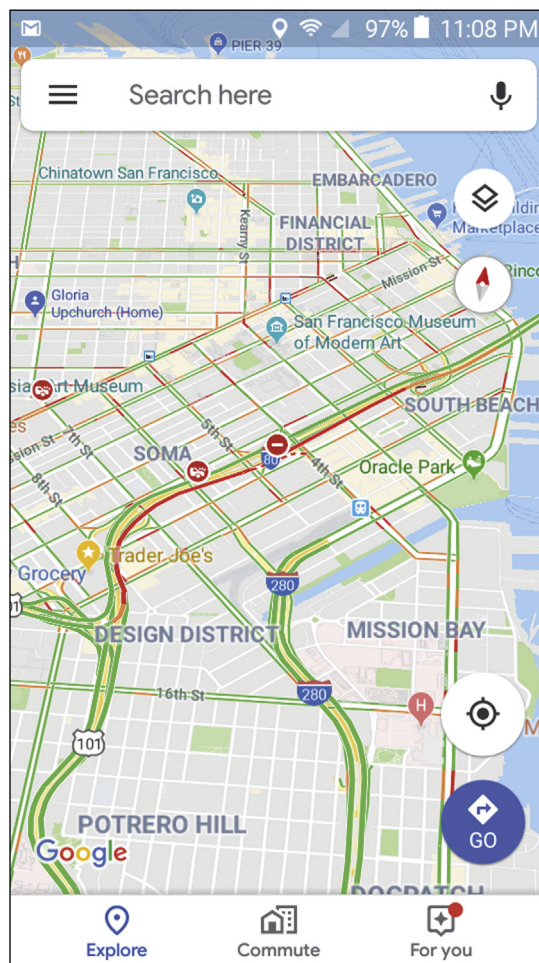


FIGURE 5.19

Google Maps uses color to show traffic conditions. Red indicates traffic jams.



FIGURE 5.20

Air traffic control systems often use red to make potential collisions “pop” out.

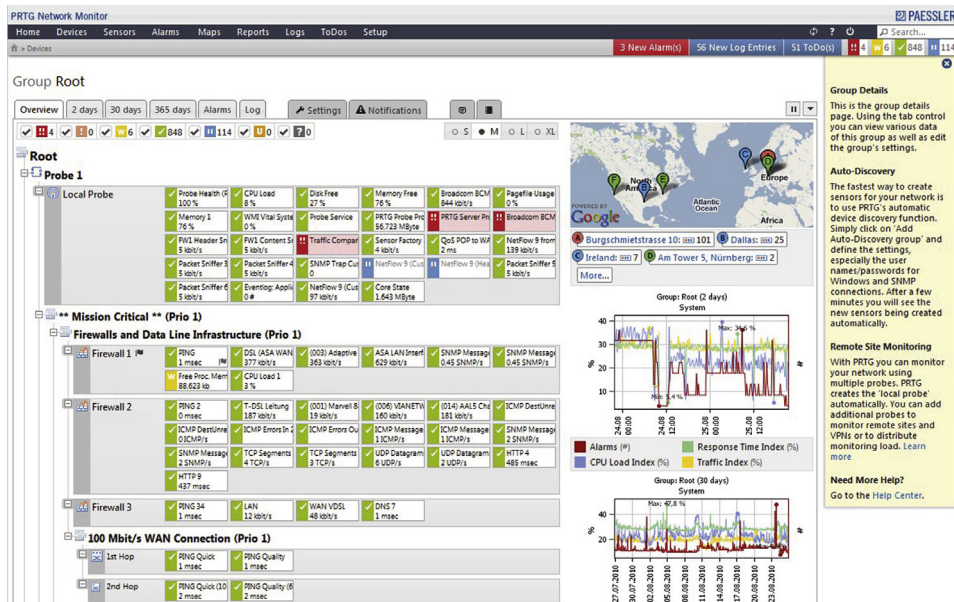


FIGURE 5.21

Paessler’s monitoring tool uses color to show the health of network components.

in red (see Fig. 5.20). Applications for monitoring servers and networks use color to show the health status of assets or groups of them (see Fig. 5.21).

These are all uses of peripheral “pop” to make important information stand out and visual search nonlinear.

When there are many possible targets

Sometimes in displays of many items, *any* of them could be what the user wants. Examples include command menus (see Fig. 5.22A) and app pallets (see Fig. 5.22B). Assume that the application cannot anticipate which item or items a user is likely to want and highlight them. That is a fair assumption for today's applications.⁵ Are users doomed to have to search linearly through such displays for the item they want?

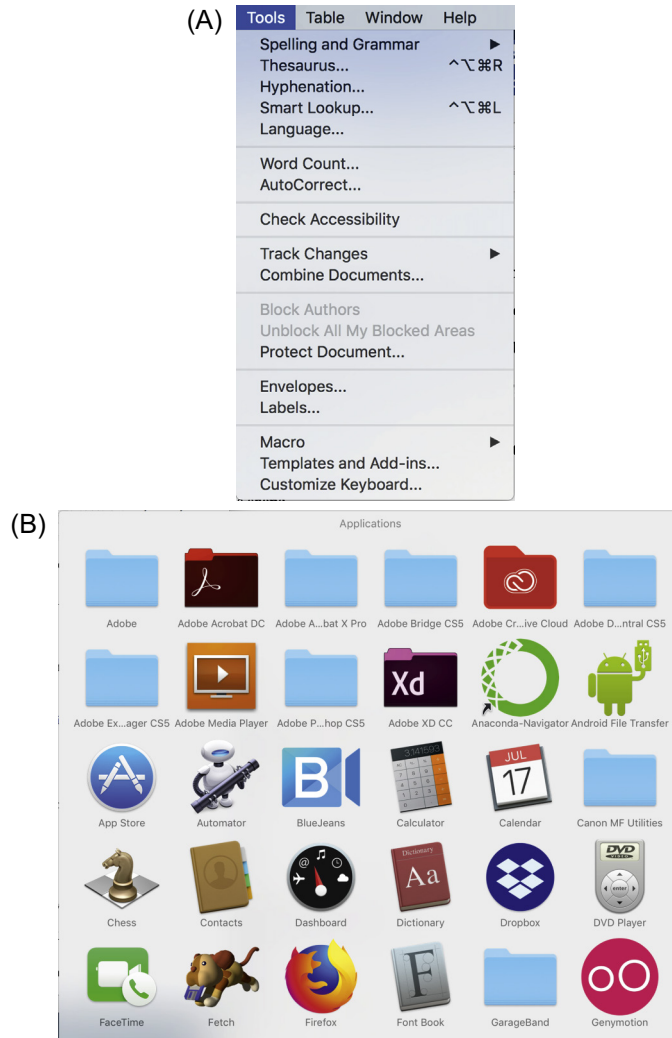


FIGURE 5.22

(A) Microsoft Word tools menu and (B) macOS application pallet.

⁵ But in the not-too-distant future it might not be.

That depends. Designers can try to make each item so distinctive that when a specific one is the user's target, the user's peripheral vision will spot it among all the other items. Designing distinctive sets of icons is hard—especially when the set is large—but it can be done (see Johnson et al., 1989). Designing sets of icons so distinctive that they can be distinguished in peripheral vision is *very* hard, but not impossible. For example, if a user goes to the macOS application pallet to open his or her calendar, a white rectangular blob in the periphery with something black in the middle is more likely to attract the user's eye than a blue circular blob (see Fig. 5.22B). The trick is to not get too fancy and detailed with the icons—give each a distinctive color and gross shape.

On the other hand, if the potential targets are all words, as in command menus (see Fig. 5.22A), visual distinctiveness is not an option. In textual menus and lists, visual search *will* be linear, at least at first. With practice, users learn the positions of frequently used items in menus, lists, and pallets, so searching for particular items is no longer linear.

That is why applications should *never* move items around in menus, lists, or pallets. Doing that prevents users from learning item positions, thereby dooming them to search linearly forever. Therefore, the use of “dynamic menus” is considered a major user-interface design mistake (Johnson, 2007).

IMPORTANT TAKEAWAYS

- Unlike digital cameras, the resolution of human vision is much higher in a small area in the middle of our visual field than it is everywhere else. The small area is called the *fovea*, and it makes up only about 1% of our visual field. Our peripheral vision has very low resolution.
- Peripheral vision guides our eyes and attention toward objects and events that either match our goals or represent possible threats. Peripheral vision can detect motion and tends to move our eyes toward whatever is moving even though it cannot identify *what* is moving.
- Peripheral vision is good in low-light situations.
- Some visual features “pop out” in peripheral vision, and some do not. Font weight pops. Color pops. Motion pops. Letter shape does not pop.
- Designing based on the strengths and weaknesses of peripheral vision:
 - Place new, important, or changed information where users will be looking—in or near where their fovea is positioned. Information placed elsewhere may not be noticed.
 - Use color, motion, a distinctive shape, etc. to make important information “pop” in peripheral vision to attract users' fovea and attention. Red is commonly used to attract attention to error messages.

- Overuse of any stimulus causes people to habituate to it, diminishing its ability to attract attention. Pop-up messages can in principle force users to attend to them, but they are often overused, and therefore many experienced users of digital technology have learned to ignore them.
- Sound can attract a user's attention, but it can also be annoying, especially in environments shared with other people.