Data Visualization | Niall Williams

August 22, 2017

Scientific	Information
Physical	Can’t distinguish the subject
Reconstruction of world	Abstract data
Spatially-embedded data (has xyz location)	

August 24, 2017

Functional Programming

In computer science, functional programming is a programming paradigm—a style of building the structure and elements of computer programs—that treats computation as the evaluation of mathematical functions and avoids changing-state and mutable data.

Core concepts:

1) Keeps functions and data separate
2) Avoids state change and mutable data
3) Treats functions as first-class citizens

Keeps functions and data separate - Functional programming stores data in simple constructs like arrays and hashes and makes separate functions that take the data as an argument and return some sort of transformed version of the data. This is done this way because it gives us immediate polymorphism in regards to the data. For example, a function that edits the age of a given object will be able to work with any object that has an age data type, instead of having to create a class for each object each with their own separate `increaseAge()` function

Avoids state change and mutable data - Data is made immutable in functional programming so that we do not have to worry about changing the state of our data as it gets called multiple times throughout the program (which can be hard to keep track of if it happens). The data is made immutable either through using constants, or by treating out data in immutable ways. Thus, we would have to create a new variable for each change we make to data. It makes debugging a lot easier. The cost of copying and then transforming this data is not very expensive because, as mentioned before, the data is stored in simple structures.

Treats functions as first-class citizens - This means we can declare functions as variables, just the same way we could store a string as a variable. Thus, first-class functions allow us to pass functions as arguments to other functions as well as return functions. This leads to much more flexibility.

```
const myShoppingList = [
  { name: "milk", price: 4.99 },
  { name: "eggs", price: 3.99 },
  { name: "beer", price: 6.99 }
]

const nextWeeksShoppingList =
  addItem(myShoppingList,
  { name: "bananas", price: 1.99 })
```

```
const monkeyList = addItem(myShoppingList,
  { name: "bananas", price: 1.99 }
const veganList = removeItems(myShoppingList,
  "milk", "eggs")

if (buyPetMonkey) {
  sendToWife(monkeyList)
} else if (guestsAreVegans) {
  sendToWife(veganList)
} else {
  sendToWife(myShoppingList)
}
```

To assign a function to a variable, simply write a function after the equals sign (and define what the function does). Ex:

```
Const greet = function(){
      console.log("Hello")
}
greet() //"Hello"
```

When functions are defined the normal way (not assigned to a variable), it doesn't matter if we call the function before or after it is defined in the code. This is because the compiler automatically moves these functions to the top of the file.
When functions are assigned to variables, we can only use them after they are declared (just like any other variable). This can be useful if we want to make code more readable by renaming multiple function calls to a more readable code.
Putting parenthesis after a function variable transforms it into whatever its return value is.

A higher-order function is a function that takes a function as an argument, returns a function, or both. Higher-order functions are useful when working with multiple functions that have similar needs and usages.

<u>Array Functions</u>
Remember to require lodash:

```
var _ = require("lodash")
_.map(ARRAY)
```

Mapping - Mapping allows you to edit all the values of an array with some function you define, and returns a new array (not the same original array) with all the values edited by the function. For example, you can use this to square every value in an array.

Filtering - Filtering allows you to create and return a new array that only contains certain elements from the original array that meet specific defined criteria. For example, this can be used to get all the even numbers from an array.
Every/some - These two functions are similar to filter, but instead of returning an array, they either return true or false. `_.some` returns true if the function we give it returns true for **at least one** of the elements in the array. `_.every` returns true only when the elements **all** return true for the given function.
Reducing - This takes an array and, based on the function we give it, reduces the array down to a single value. For example, it might take an array and reduce it down to a sum or average of the elements. The value we pass into the `reduce` function requires 2 arguments. The first is the value accumulated so far, and the second is the current element. Each time the function is called, its return value becomes the first argument for the next time it's called. If we pass in a third argument to `reduce`, this argument is the starting value of the function.
Combining functions - We can combine the functions above to do a lot more. For example, you can filter an array of employees to calculate the average salary of all male employees and then do the same for female employees and then compare the values.

Callbacks

This allows for asynchronous function calls, meaning we can set delays to when a function is executed so that the program is able to compute and execute other functions while the asynchronous function is still being computed. Keep in mind that asynchronous functions can be nested to allow for even more specific ordering.
Callbacks with arguments - It is standard to pass the result of the function as the second argument, and any errors as the first argument.

Partial Application

This allows you to only apply a function once we have all the information (arguments) ready.

Recursion

You already know this stuff… yay

August 28, 2017

The advantages of a visualization include:

- An ability to comprehend a huge amount of data very quickly (think heatmap)
- Allow the perception of emergent properties that were not immediately apparent/obvious before.
- Often highlights problems or outlying oddities in the dataset.
- Facilitates understanding of both large-scale and small-scale features of data.
- Facilitates hypothesis formation.

The stages of visualization are:

1) The collection and storage of data
2) Preprocessing stage designed to transform data into something usable and easy to manipulate.
3) Mapping the selected data to a visual representation (using a computer, usually)
4) Human perceptual and cognitive system (the perceiver)

<u>Semiotics of Graphics</u>

Semiotics is the study of symbols and how they convey meaning. The ability of humans to interpret images (such as pictures, paintings, and drawings) is an innate ability. Humans do not need to be taught that, for example, a drawing of a dog is meant to represent a dog; they are able to understand that the drawing is a representation of a dog (in the sense that they can identify what object the artist was intending to draw). Basically this means that the symbols we choose to represent data should be picked carefully - we can't just pick any random symbols and define them and then expect another visualization to do the same thing and be just as good. One will probably be a better visualization based on how the body interprets symbols.

Sensory refers to symbols and aspects of visualizations that get their expressive power not from learned/taught origins, but rather are innately able to take advantage of the way the human body perceives stimuli in order to convey a message.

Arbitrary refers to symbols that have no inherent meaning or power, and are simply used in a certain way because that is the definition humans have given to it.

Sensory visualizations are usually stable across cultures, individuals, and time. However, arbitrary conventions derive their power from culture and are thus more dependent on the perceiver.

<u>Properties of Sensory Representation</u>

- **Understanding without training** - A sensory code is one for which the meaning is perceived without additional training or learning.
- **Resistance to alternative denotation** - Certain aspects of symbols and visualizations cannot be ignored. For example, the effects of optical illusions persist after the fact that we learn that they are optical illusions. Likewise, the use of lines connecting 2 objects when we do not want to indicate a connection between the objects is **bad**.
- **Sensory immediacy** - Certain methods of representation and visualization are perceived more easily than others. For example, some textures are easier to distinguish between vs other sets of textures.
- **Cross-cultural validity** - Sensory codes will generally be understood across cultural boundaries.

Based on these rules, we can arrive at the first visualization guidelines, tagged **[GUIDE #.#]**.

[GUIDE 1.1]
Design graphic representations of data by taking into account human sensory capabilities in such a way that important data elements and data patterns can be quickly perceived.

[GUIDE 1.2]
Important data should be represented by graphical elements that are more visually distinct than those representing less important information.

[GUIDE 1.3]
Greater numerical quantities should be represented by more distinct graphical elements.

Sometimes, guide points will conflict with each other. In these cases, it is up to the designer to best determine how to represent their information to convey the most important messages.

Properties of Arbitrary Representation

The development of arbitrary representations (such as number systems) have occurred over only the most recent few thousand years, as opposed to the millions of years it has taken for humans to develop sensory representation systems. Computer graphics have greatly advanced our ability to create new representation codes. The properties of arbitrary representations are as follows:

- **Hard to learn** - The defined rules of any arbitrary system must be studied and adhered to at all times in order for these systems to work effectively. This can take considerable amounts of time (such as the case of learning a new language).
- **Easy to forget** - Defined rules and meanings that have no correlation to the ideas/objects they define are easy to forget because it is not intuitive. It is also the case that arbitrary codes can interfere with each other, which is not the case for sensory codes.
- **Embedded in culture and applications** - Different cultures have created their own arbitrary codes for systems that can be, and often are, wildly different from the codes of another culture for the same system.
- **Formally powerful** - Arbitrary graphical notations can be constructed such that they embody formally defined, powerful languages. For example, mathematicians have created hundreds of graphical languages to express and communicate their concepts.

This leads us to our fourth guideline below:

[GUIDE 1.4]
Graphical symbol systems should be standardized within and across applications.

<u>August 29, 2017 (class)</u>

Remember that there is always a user - there will always be someone trying to interpret your data. Take this into account when creating a visualization.

A good visualization lets us see patterns in the data

Visual Processing

- Early visual processing - extracts of linear features. Pictorial outline and image itself excite same neural processes

The brain automatically seeks out and identifies lines within images, whether they are distinctly intended or not.

If you can make use of sensory images, you probably should. It matches the first stages of neural processing.

<u>August 30, 2017</u>

<u>Gibson's Affordance theory</u>

Gibson argued that we perceive in order to operate on the environment; perception is designed for action. He called the perceivable possibilities for action **affordances**, and a major point of his theory is that affordances are perceived in a **direct** and immediate way. Thus, they are **not inferred** from sensory clues. He believed that instead of perceiving points of light, we instead perceive the possibilities for action, such as surfaces for walking, handles for pulling, etc.

However, there are a few problems with Gibson's approach. The **first problem** is that even if perception of the environment is direct, it's clear that a visualization of data via computer graphics is indirect (there are many layers of processing between the data and the visualization). Direct perception is not always applicable, such as in the case of when the data comes from a microscopic source.

The **second problem** is that there are no clear physical affordances in any GUIs. One cannot say that a button on the screen "affords" clicking it in the same way some surface affords walking on it. We need to know that buttons on the screen can even be pressed, which is not a direct interaction with the physical world.

The **third problem** is that Gibson rejects visual mechanisms. Gibson rejects the importance of understanding perceptual mechanics, which in turn rejects most of vision research as irrelevant.

<u>Model of Perceptual Processing</u>

There are three main steps involved in the human perception process. They will be discussed on the **next page.**

Stage 1: Parallel Processing to Extract Low-Level Properties of the Visual Scene

This first step involves billions of neurons all extracting features from every part of the visual field simultaneously. Some neurons are specifically tuned to detect certain features, such as edges. It occurs whether we want it to or not, and is very fast. If we want some information to be easily and quickly detected by the perceiver, it should aim to be detected by these large, fast computational systems. Important characteristics of stage 1 processing include:

- Rapid parallel processing
- Extraction of features/orientation/color/texture/movement patterns
- Transitory nature of information
- Bottom-up, data-driven model of processing
- Serving as the basis for understanding the visual salience of elements in displays

Stage 2: Pattern Perception

During this stage the visual field is divided into regions and simple patterns (contours, regions of the same color/texture, etc). Important characteristics of this process include:

- Slower serial processing
- Top-down attention being critical to the formation of objects and patterns pulled out from the feature maps
- A small number (1-3) of patterns becoming bound and held for a few seconds under top-down attentional processes
- Different pathways for object recognition and visually guided hand motion

The pattern-processing pathway includes two major processes. One such pathway deals with object perception and the other deals with parts of the brain involved in the control of actions. It is the basis for the two-visual-system theory: one system for locomotion and action (**action system**) and one for object identification (**what system**).

Stage 3: Visual Working Memory

In the highest level of perception, the objects are held in visual working memory by the demands of active attention. Here, only a few objects can be held at any time (constructed from the available patterns).

Attention describes the way in which the visual system is being tuned in accordance to what we are seeing and what we expect to see. Attention is most prominent in stage 2, which is where attention decides where to move our eyes in order to find objects we are looking for (focus of our attention).

Costs and Benefits of Visualization

The main goal of creating visualizations is to assist in work in some form so that cognitive tasks can be completed more efficiently. A visualization has two main perspectives, the user and the developer. The **basic user cost** is as follows:

*(time to learn to use the visualization * value of the user's time) + (time spent carrying out the work * value of the user's time)*

The **user benefits** are as follows:

*cognitive work done * value of the work*

This leads to another guidelines:

[GUIDE 1.5]

Where two or more tools can perform the same task, choose the one that allows for the most valuable work to be done per unit time

[GUIDE 1.6]

Consider adopting novel design solutions only when the estimated payoff is substantially greater than the cost of learning to use them

Sometimes, learning to interpret a novel data representation can take too much time that could be spent on something else. Often, it is not worth learning a new tool, **especially if the number of times it will be used is uncertain**.

[GUIDE 1.7]

Unless the benefit of novelty outweighs the cost of inconsistency, adopt tools that are consistent with other commonly used tools.

Now, we will look at the developer side of visualizations. The **basic developer costs** are as follows:

cost to design and implement a cognitive tool + cost to market + cost to manufacture + cost to service

The **developer benefits** are:

*(number of units sold * price per unit) + revenue from maintenance contracts*

[GUIDE 1.8]

Effort spent on developing tools should be in proportion to the profits they are expected to generate. This means that small-market custom solutions should be developed on for high-value cognitive work.

Types of Data

It is important to be able to classify types of data so that we can apply these data types to the various visualizations we are doing. If we can give perceptual reasons for classifying data types as being good for certain types of visualizations, then we have an applied science of visualization. However, classifying types of data is quite hard. Contained in this section will be informal data classifications.

The most basic classification is dividing data into **entities** and **relationships (relations)**. **Entities are the objects we wish to visualize. Relations define the structures and patterns that relate entities to one another.**

Entities are usually the objects of interest.

Relationships form the structures that connect entities. There can be multiple kinds of relationships for different (or the same) entities. Relationships can be structural/physical, or they can be conceptual. They can also be causal (one event causes another) or they could be purely temporal (defining an interval between two elements).

Entities and relationships can both have **attributes**. Something should be called an attribute (instead of calling it an entity) when it is a property of some entity and **cannot be thought of independently**. For example, the redness of an apple is an attribute of the apple.

Types of Numbers

Often times a visualization will incorporate values to help convey a message. There are 4 types of numbers:

- **Nominal** - This is a labeling number. Something like the numbers on a license plate - they exist only to serve the purpose of identifying certain objects.
- **Ordinal** - Ordinal numbers are used for ordering things in a sequence, including things such as rankings or order in a queue.
- **Interval** - These numbers make it possible to derive the gap between data values. One example of this kind is the time of departure and arrival of a car.
- **Ratio** - A ratio scale utilizes the full expressive power of a number. It allows us to compare objects to each other by some attribute (such as size). The use of a ratio scale implies a zero value used as a reference.

Additionally, there are types of data that are most often considered in visualizations. They make use of the types of numbers mentioned above. They are:

- **Category data** - This includes things like the nominal values
- **Integer data** - Includes the ordinal class (it is discrete and ordered)
- **Real-number data** - This combines properties of interval and ratio scales

An important aspect of data to note is the process of transforming data. This can include anything from adding two groups of values (data) to inverting values to yield opposites to forming a new object out of other objects. When data gets transformed, it is often difficult to communicate this transformation in a visualization, so this must be taken into account when making a visualization. Animations can help a lot in this regard.

August 31, 2017 (class)

Class material continued. Remember that color has different meanings in different cultures. Remember that human eyes follow curves/curved lines much easier than right angled lines.

Tuesday quiz 1: Guidelines 1.1-1.4. Sensory and arbitrary symbols. Written and in class. No coding section. There will be code on the quiz, so know how to read code and fill in the blanks/know the output of code. Understand and discuss functional programming. Pages 1 - 17. Any in-class information and slides. 45 minute quiz. Know and explain basic definitions and guidelines.

September 5, 2017 (class)

Gibson and perceptual processing

Gibson said:

- We perceive in order to operate on the environment
- Perception is designed for action
- **Affordances - the perceivable possibilities for action**

In short, he argued perception is designed for action.

Problems with Gibson's theory are:

- Visualization of data through a computer is indirect while the world is direct.
- Are there actually affordances in GUIs? How can we necessarily know that a button can be pushed?
- Gibson rejected the bottom-up approach to vision, meaning he thought we perceive first, then see an action, rather than trying to understand the intricacies of how vision works. This means you may create confusing visualizations (optical illusions) if you don't understand how visual processing works.

Perceptual processing

There are 3 stages to visual processing:

1) **Features** - We extract features from what we look at, which are color, orientation (of lines), texture, and motion. They should be obvious/jump out at you. This stage involves rapid parallel processing. This stage is under the bottom-up processing. We see the objects/features, and we will later put them into the bigger picture.

2) **Pattern perception** - Slower than stage 1. Humans see only a few patterns at once, and will ignore others until they have been processed. Involves top-down attention.
3) **Visual working memory** - This is where everything from steps 1 and 2 get put together so that we can begin answering specific questions and arrive at specific conclusions. Can be thought of as the critical thinking process, almost.

September 11, 2017

The Visual Angle Defined

The visual angle is key in defining the properties of the eye and early vision. It is defined as **the angle subtended by an object at the eye of an observer.** They are defined in degrees, minutes, and seconds of arc (A minute is 1/60 degree and a second is 1/60 minute). A thumb nail held at arm's length subtends about 1 degree of visual angle. To calculate visual angle, use the equation:

$$tan(\theta/2) = 2/(2d)$$
$$\theta = 2\ arctan(h/2d)$$

Lens

The human eye contains a compound lens, which has 2 important elements: the curved front surface of the cornea and the crystalline lens. The following equation gives the imaging properties of a simple lens:

$$1/f = 1/d + 1/r$$

Where f is the focal length of the lens, d is the distance to the object that is imaged, and r is the distance to the image that is formed.

If the units are meters, the **power** of a lens is given by the reciprocal of the focal length (1/*f*) in units of **diopters**. Thus, a 1-diopter lens has a focal length of 1m.

In the eye, the majority of the compound lens' power comes from the front surface of the cornea (about 40 diopters), while the rest comes from the variable-focus lens. The **depth of focus of a lens is the range over which objects are in focus when the eye is adjusted for a particular distance.** For the human eye, this varies with the size of the pupil, but it is safe to assume that in most cases, objects that are between 3m and infinity distance away are in focus.

Chromatic Aberration

The human eye is not corrected for chromatic aberration, meaning that different wavelengths of light are **not** focused at different distances within the eye (so, chromatic aberration means that different wavelengths of light are focused at different distances within the eye). So, a blue and a red light next to each other require different levels of focus to avoid blur when looking at either one individually, since they have different wavelengths. This means if you look at red objects next to blue ones, the blue ones will be blurry when you focus on the red.

Because of chromatic aberration, one should avoid making fine patterns that use undiluted blue phosphor on a black background (??? ok..). Adding a small border of green or red to the blue object will force the blue to no longer be blurry, because the green/red provide luminance edges to define the blue boundaries. This lack of chromatic aberration can lead to powerful depth optical illusions.

<u>Receptors</u>

The retina has 2 cell types: rods and cones. Rods work well in low light, and cones work well under normal light levels. Because of this, rods are usually disabled throughout the day (too much light in the daytime), so they are usually ignored when considering visualization specifics. In the middle of the retina is the **fovea** which is densely packed with cones, and is where vision is the sharpest.

<u>Simple Acuities</u>

Visual acuities are measurements of our ability to see detail. They tell us the limits on the information densities that we can perceive. In general, we can resolve things down to about 1 minute of arc. For us to tell 2 objects apart, the space between them should lie on a receptor, and therefore we should only be able to perceive lines separated by roughly twice the receptor spacing.

There are some exceptions to this rule, known as superacuities. **A superacuity is the ability to perceive visual properties of the world to a greater precision than could be achieved based on a simple receptor model.** A good example of this is the vernier acuity, which involves accuracy better than 10 seconds of arc. So, since a computer monitor usually has 40 pixels per centimeter, we can perform vernier acuity tasks that are accurate to about 1/10th of a pixel.

Our brains are able to integrate information over space and time to allow for perception of higher resolution information than is actually available on our display device. One way of doing this is through anti aliasing.

<u>Acuity Distribution and the Visual Field</u>

This refers to field of vision, pretty much. I have a feeling this is not a very important section, so I didn't write anything about it. Maybe it will be important. It's on page 52-55, though.

<u>**September 12, 2017 (class)**</u>

Weather info stuff. Assignment 2

Are the storm events synchronous between streams? Ie start and stop at the same time in each stream

- Relationship between temperature and discharge (Q)

- Conductivity data? Is it even useful? It's very messy. Is there a relationship between the time intervals and the fluctuations of the conductivity?
- Wall 272

<u>September 14, 2017 (class)</u>

Calculating visual angle is probably on the next quiz.

32 degree - 13 sec

<u>Acuity</u>

Measurement of our ability to see detail.

a) Point (1 min of arc)
b) Grating (1-2 min of arc)
c) Letter (5 min of arc)
d) Stereo (10 min of arc) (not covered in this class. Applies to 3D/VR stuff)
e) Vernier (10 min of arc)

Rods are bad at seeing color, but good at seeing motion. Optic flow is the term for what you see in your periphery.

[GUIDE 2.1]

Red should not be a background color (since red tends to jump out). Do not make fine patterns of blue on black (hard to read). Red and white attract the focus. Blue is a good background color in general.

Luminance is the intensity of light emitted from a surface per unit area in a given direction. Luminance is **measured**. It is unrelated to perceived brightness and perceived lightness.

<u>Receptive Field</u>

Where cell responds to light. (retinal ganglion cells). They are basically light detectors.

Note: **It is impossible to perceive that luminance values are equal, even if you know it. This is because the surrounding objects/colors affect how you perceive it.**

What we can take away from this info:

- **Grayscale color map** - Errors averaging 20% of the entire scale when trying to make observations regarding differentiating between absolute values. I.e. people are bad at seeing differences in grayscale if there are too many shades of gray being used.
- **Edge enhancement** - You can use luminance shading to define borders and emphasize edges.

- **Luminance perception** - Luminance is completely unrelated to perceived brightness or lightness.

You must have luminance changes in order to perceive depth in 2D images.

[GUIDE 2.2]
Never use a gradient background, EVER.

<u>**September 19, 2017 (class)**</u>

If you have some outliers and you have a color scale, you can set the outliers to their own separate color on the scale so that the non-outliers actually utilize the scale and have color differences, instead of having the outlier polarize the scale and clump all the non-outliers into one color scale. Ex: [min, 90%, max].

<u>Luminance, Brightness, Lightness</u>

Luminance is very essential to determining the depth of objects. It is completely unrelated to chrominance. Luminance proves the shape of the object. **It is the measured amount of light. It is the basis of our pattern, shape, and motion perceptions.**

If you are sorting by black and white, you can only have up to 4 categories.

[GUIDE 3.1]
Avoid using gray scale as a method for representing more than a few (two to four) numerical values)

Quick changes in luminance are very apparent. Be very careful with color. Color is an extra feature, but luminance allows you to actually tell what is going on in the data trends.

[GUIDE 3.3]
Consider using adjustments in luminance contrast as a highlighting method

<u>Pattern Perception</u>

Color change will **not** help you see pattern. Note: colors have measured luminance, so the luminance channel of the color might affect how you see the patterns. It’s not the color itself, but rather the luminance.

[GUIDE 3.4]
Use a minimum 3:1 Luminance contrast ratio between a pattern and its background whenever information is represented using fine detail, such as texture variation, small-scale patterns, or text.
10:1 for text is recommended by the International Standards Organization

Use http://springmeier.org/www/contrastcalculator/index.php to check your ratios

Note: The finer the detail, the greater the luminance contrast is required.

Motion Perception
Luminance change is needed to perceive motion.

Brightness: Perceived amount of light
Lightness: Perceived reflectance of a surface

RGB values should all be the same in order for us to be on the grey scale. Moving all the RGB values closer together will grey out the color.

September 26, 2017 (class)
http://colorbrewer2.org/#type=sequential&scheme=BuGn&n=3
https://github.com/d3/d3-scale-chromatic
Note: Color for categories of data is good, but only if you don't have too many categories.

Data types

- **Nominal** - Naming/categorical. For data that is not ordered in any way. Ex: types of fruit.
- **Ordinal** - Ordered data. Has a 1st, 2nd, 3rd, …, order. However, you don't know what the quantitative difference between items is. So, if 1st item comes 10 years before the 2nd item, it would be the same as if 1st item came 1 minute before the 2nd one. Ex: Class years. There are 5th year seniors, or juniors that are about to graduate.
- **Ratio** - Real numbers.

Quiz 2:
Same format. Everything from last quiz basically/ Coding questions included. Color maps, luminance, color aberrations.

September 28, 2017
Multivariate Visualization

Univariate visualization

- **Single variable** - Bar chart, line graph, box plot, etc. Often has a numerical and categorical axis.

Bivariate visualization

- **Scatter plot** - show clusters and correlation. Compares 2 variables to show a connection between them.

Multivariate visualization:

Involves 2 basic strategies:

- **Dimension Reduction** - use mathematical equations to reduce our data. Ex: project a 3D point onto a 2D plane. Principle component analysis (PCA) is when you plug your data into something like R and it maximizes the variance to find orthogonal axes. You usually pick the 2 highest orthogonal axes that you care about and plot them onto a lower dimensional representation. Always involves losing some info. Multidimensional Scaling (MDS) is when you position points in lower-dimensional space to closely match distance in a high-dimensional space. It is heuristic and thus may not produce the same layout or the optimal layout (?? figure this out). Another is Self-Organizing Maps (SOM), which uses neural networks and uses a random grid to move similar items together. A topological/geometric approach can allow you to map things into 3 dimensions with shapes.
- **Direct Visualization** - This involves trying to directly represent all (or a high dimension subset) of the data dimensions. You map multiple dimensions to visually salient features. Involves **glyphs**, which is a single graphical icon displaying multiple variables, such as position, shape, color, size, orientation, etc. The way it's mapped is task-dependent. Integral vs separable dimensions and preattentive vs attentive dimensions. Fortunate to get about 8 dimensions (you can use dimension reduction if you don't need them all). The layout using positioning can cause overdrawing (occlusion. Z dimension).

Parallel Coordinates - Each dimension on a parallel axis, and then you draw lines between points to connect the data. You can use color for the categories the make them easier to read, or use brushing. You can add transparencies which allows you to see connections since the connected data is all on top of each other. You can curve the lines to follow them more easily (no sharp angles). You can blur out sections of the lines to see the more interesting points. You can even extend them into 3D.

October 11, 2017

Color

While color does not help us determine the spatial arrangement of objects, how they are moving, or what shapes they are, color is still very useful. It allows us to break camouflage. Some things are different from their surroundings only because of their color. Additionally, color tells us about the material properties of objects, which is very important for judging the condition of our food, for example. Think of color as an attribute of an object, rather than a primary characteristic.

Color is good for labeling and categorization, but poor for displaying shape, detail, or spatial layout.

Trichromacy Theory

In our eyes we have 3 different kinds of cones, which is the reason for basic three-dimensionality of human color vision. **Color space is an arrangement of colors in a three-dimensional space.** You can convert from one color space to another with some calculations. We can match a particular patch of colored light with a mixture of just 3 colored lights, called **primaries.**

Color blindness - One downside of using colors to display information is that some of our audience may be color blind. About 10% of males and 1% of females are color blind. Most commonly, it is because of a lack of long-wavelength-sensitive cones (protanopia) or the medium-wavelength-sensitive cones (deuteranopia). Both of these result in an inability to distinguish between red and green. For color blind people, the 3 dimensional color space basically collapses to a 2 dimensional space.

Color Measurement

We can describe a color with the following equation:

$$C \equiv rR + gG = bB$$

Where C is the color to be matched; R, G, and B are the primary light sources to be used to create a match; and r, g, and b represent the amounts of each primary light. The ≡ symbol is used to denote a perceptual match. (looks the same to humans).

The *RGB* primaries form the coordinates of a color space. If these primaries are physically formed by the phosphor colors of a color monitor, this space defines the **gamut** of the monitor. **Gamut is the set of all colors that can be produced by a device or sensed by a receptor system.** The relationship defined in the equation above is a linear relationship. If we double the amount of light on the left, we can double the amount of light on each of our primaries and the match will still hold. We can also introduce negative values into the equation. A negative value basically means we move the value of the left side, so its light is added to the light of the sample. When we do this, it means we can have any colored light matched by a weighted sum of *any* three primaries as long as each is distinctive in cone space. The primaries do not even have to match an actual color, and in fact the most widely used color standard is based on non-physical primaries.

Change of Primaries

The use of red, green, and blue for the primaries is arbitrary. We can change from one set of primaries to another if we want. Given a standard way of specifying colors (using a standard set of primaries), we can use a transformation to create that same color on any number of different output devices.

One of the basic concepts in any color standard is that of the standard observer. This is a hypothetical person whose color sensitivity functions are held to be the standard for all humans. It assumes that we all have the same receptor functions.

Most serious color specification is done using the **Commission Internationale de l'Eclairage (CIE)** system of color standards. The CIE system uses abstract observer sensitivity functions called **tristimulus values**, which can be thought of as a set of abstract receptors and they are labeled *XYZ*. Note that the *Y* tristimulus value is the same as luminance.

<u>Chromaticity Coordinates</u>

This is a measurement that defines the hue and vividness of a color while ignoring the amount of light. There are certain properties of a chromaticity diagram that are important:

1) If two colored lights are represented by two points in a chromaticity diagram, the color of a mixture of those two lights will always lie on a straight line between those two points.
2) Any set of three lights specifies a triangle in the chromaticity diagram. Its corners are given by the chromaticity coordinates of the three lights. Any color within that triangle can be created with a suitable mixture of the three lights.
3) The **spectrum locus** is the set of chromaticity coordinates of pure monochromatic (single-wavelength) lights. **All realizable colors fall within the spectrum locus.**
4) The **purple boundary** is the straight line connecting the chromaticity coordinates of the longest visible wavelength of red light (about 700 nm) to the chromaticity coordinates of the shortest visible wavelength of blue (about 400 nm).
5) The chromaticity coordinates of equal-energy white (light having an equal mixture of all wavelengths) are 0.333, 0.333. (And some other crazy shit on page 104 that I don't think we need to know).
6) **Excitation purity** is a measure of the distance along a line between a particular pure spectral wavelength and the white point. Another term for this value is the **saturation**. More saturated colors are more vivid.
7) The complementary wavelength of a color is produced by drawing a line between that color and white and extrapolating to the opposite spectrum locus. Adding a color and its complementary color produces white.

A widely used standard for the color of monitor primaries is *sRGB***.**

<u>Color Differences and Uniform Color Spaces</u>

Sometimes, we want equal perceptual distances to have equal distances in the color space. They are useful in the following situations:

- **Specification of color tolerances** - Useful for when manufacturers wish to order colored parts from a supplier, and need to specify a color tolerance within which each part will be accepted
- **Specification of color codes** - If we need colors to code data, such as the different wires in a cable, we want these colors to be as distinct as possible to avoid confusion. We can do this by making them as far apart as possible in a uniform color space.
- **Pseudocolor sequences for maps** - Many scientific maps use **pseudocoloring** to represent ordered data values with sequences of colors. A uniform color space allows us to create perceptually equal steps in a sequence of colors.

Though they are useful, uniform color spaces only provide a rough first approximation of how color differences will be perceived. There are a lot more factors that influence color perception in complex environments, such as contrast or different sized patches of light

[GUIDE 4.1]

Use more saturated colors when color coding small symbols, thin lines, or other small areas. Use less saturated colors for coding large areas.

<u>October 12, 2017 (class)</u>

<u>Color</u>

We **do not** need color to determine shape, spatial orientation, or motion.
We **do** need color to break camouflage. It allows us to see certain objects more easily, if it is a similar shape to its surroundings. It allows us to figure out if food is ripe or spoiled, and helps us determine what material types objects are.

Why do we want to use color in a visualization?

1) Useful for labeling and categorizing data
2) Mimics reality
3) Attracts attention to points of interest in a visualization
4) Shows groupings of data

Additionally, it looks nice and can help us emphasize patterns.

If we are using colors to categorize data, we need to know how many colors we can use. The distinct colors we can use are:

- Black
- White
- Red
- Yellow
- Green

- Blue
- Brown
- Pink
- Purple
- Orange
- Gray

We have a harder time distinguishing between different shades of blue because blue is on a low wavelength.

The magic number of max amount of colors that you should use is 8. There are 8 hues accepted as different enough to be placed into their own color categories. They are on the slide with the triangle diagram. Look at it.

If you are dealing with a nominal data set, you need to limit yourself to 10 categories or fewer.

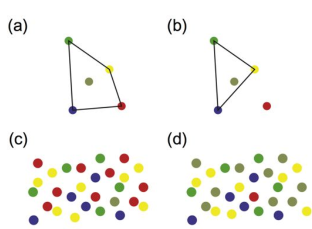
Hue - The actual color being discussed
Saturation - How "white" the color is. It stays within the same hue. 0% saturation is white
Brightness - From light to dark. How bright the color is

Remember:
4 categories for **gray scale**
8 categories for **colors**
Leave luminance channel for shape perception

<u>Color blindness</u>
10% of males are colorblind. 1% of females are colorblind.
There is a small population of females that are tetrachromats. They have 4 color receptors. They are more likely to have male children that are colorblind.

<u>Issues With Color</u>

- **Background contrast** - This is the issue of luminance values for colors. Different colors will show up differently (Easier or harder to identify) on different backgrounds. The slide specifies the correct way to choose colors for color labeling.
- **Distinctness** - This is an issue of distinctness. A color that is within the range of colors yielded from mixing colors is very hard to distinguish when placed among the colors that are being mixed. Colors outside of that mixing range are very distinct and will jump out easily.



- **Limit to the number of labels** - As explained above. Only 8 labels
- **Unique hues** - See slide

October 23, 2017

Applications of Color in Visualizations

Application 1: Color Specification Interfaces and Color Spaces

Often times, we want to let users specify their own colors. To do this, the user could be given a set of controls to specify a point in the three-dimensional color space, a set of color names to choose from, or a palette of predefined samples.

- **Color Spaces** - The simplest color interface to implement on a computer gives the user controls to adjust the amount of red, green, and blue light that combine to make a color on a monitor. The controls can be sliders, or just text fields for the 3 values. Even though this is pretty simple, people can still find it confusing if they don't understand color theory.

  Most color interfaces are based on the hue, saturation, and value (HSV) color space. ***Hue* is an approximation to the visible spectrum by interpolating in sequence from red to yellow (= red + green) to green to cyan (= green + blue) to blue to purple (= blue + red) and black to red. *Saturation* is the distance from neutral monitor values, on the white-gray-black axis, to the purest hue possible given the limits of monitor primaries. *Value* is the name given to the black-white axis.**

  It is important to separate a luminance dimension from the chromatic dimensions in a color specification interface.

[GUIDE 4.8]

In an interface for specifying colors, consider laying out the red-green and yellow-blue channel information on a plane. Use a separate control for specifying the dark-light dimension.

[GUIDE 4.9]

In an interface for designing visualization color schemes, consider providing a method for showing colors against different backgrounds.

- **Naming Color Systems** - There are very few widely agreed upon color names and color memory of humans is poor, so using a system that allows users to pick colors by names is not useful except for the simplest applications.
- **Color Palettes** - When the user wishes to use only a small set of standardized colors, a color palette is a good solution to the color selection problem mentioned above. You should let the user be able to develop a personal palette so that there is consistency across

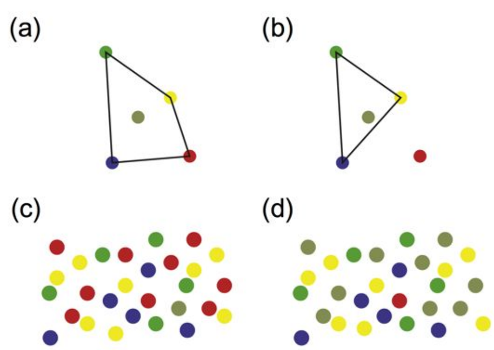
a number of visualization displays. Sometimes, the standard color sets are determined by the medium being used (such as paint or fabric), so the monitors must be calibrated to the colors appear as they actually would.

[GUIDE 4.10]
To support the use of easy-to-remember and consistent color codes, consider providing color palettes for designers.

Application 2: Color for Labeling (Nominal Codes)

Nominal information coding is when we use colored symbols to represent different groups in the dataset. A nominal code does not have to have an order, but it must be remembered and recognized. Color is the best way to distinguish between the various groups in our datasets. When choosing a set of color labels for the nominal code, there are a number of factors that must be considered:

- **Distinctness** - We need to pick colors that are distinct so that they can be easily identified and differentiated, so that the audience can efficiently understand the visualization groupings. We need to factor in different things such as background colors, symbols sizes, and application-specific requirements when picking a color code.When we want specific groups to be more distinct than others, these specific groups should **lie outside the convex hull of the surrounding colors.**



- **Unique Hues** - The unique hues (red, green, yellow, blue, black, white) are special in terms of the opponent process model. They are good colors to use, basically.

[GUIDE 4.11]
Consider using red, green, yellow, and blue to color code small symbols

- **Contrast With Background** - Another important factor to consider is the background on which symbols will appear. Backgrounds can cause some colors to appear like other colors, which we want to avoid. We can avoid this by placing a thin black or white border around all the symbols. **There should also be significant luminance differences between the symbol and the background.** Scroll down to see the guide for this point.

[GUIDE 4.12]

For small color-coded symbols, ensure luminance contrast with the background as well as large chromatic differences with the background.

[GUIDE 4.13]

If colored symbols may be nearly isoluminant against parts of the background, add a border having a highly contrasting luminance value to the colored symbol.

- **Color Blindness** - There is a substantial color-blind population, so you might want to consider using colors that can be distinguished by color-blind people.

[GUIDE 4.14]

To create a set of symbol colors that can be distinguished by most color-blind people, ensure variation in the yellow-blue direction.

- **Number** - Despite the fact that color coding is very good for displaying category information, only a small number of codes can be rapidly perceived.

[GUIDE 4.15]

Do not use more than 10 colors for coding symbols if reliable identification is required, especially if the symbols are to be used against a variety of backgrounds.

- **Field Size** - Do not use very small color-coded areas, because it makes it too hard to distinguish between symbols. The larger the area that is color coded, the more easily colors can be distinguished. When the area being coded is small, you should use strong, highly saturated colors for maximum discrimination (Guide 4.1). When the area is large, you should use less saturated colors (Like a map visualization). This allows small, vivid color-coded targets to be perceived against background regions.

[GUIDE 4.16]

Use low-saturation colors to color code large areas. Generally, light colors will be best because there is more room in color space in the high-lightness region than in the low-lightness region.

[GUIDE 4.17]

When color coding large background areas overlaid with small colored symbols, consider using all low-saturation, high-value (pastel) colors for the background, together with high-saturation symbols on the foreground.

[GUIDE 4.18]

When highlighting text by changing the color of the font, it is important to maintain luminance contrast with the background. With a white background, high-saturation dark colors must be used to change the font color. Alternatively, when changing the background color, low-saturation light colors should be used if the text is black on white.

- **Conventions** - Color-coding conventions must sometimes be taken into account. Some of these may include red = hot, red = danger, blue = cold, green = life, or green = go. You must also remember that these conventions **may not always pervade cultural borders**.

The following is a list of 12 colors recommended for use in color-coding: **red, green, yellow, blue, black, white, pink, cyan, gray, orange, brown, purple.** These are chosen because they are generally agreed to be distinct colors that are reasonably far apart in the color space.

Application 3: Color Sequences for Data Maps

Pseudocoloring is the technique of representing continuously varying map values using a sequence of colors. The result is called a choropleth map.

The most common coding scheme used in data visualization is a color sequence that approximates the physical spectrum. This spectrum is pretty good for physics and other sciences, but it is not a perceptual sequence, so there is no inherent ordering to the colors. This basically means you lose a dimension of information if you choose to encode the ordered data in colors (since they are not perceptually sequential).

- **Form and Quantity** - Sometimes we want to see the forms in a dataset. Where are the highs and lows, the ridges and spirals, etc? Color theory says that the different color schemes have very different effects in this regard. The luminance channel helps us see forms, so a **grayscale sequence should allow us to see forms much better than pure color sequences with no luminance variations**. Despite this, there are some advantages to using color sequences here. It results in much lower errors in reading values from a key. There are also sometimes semantic reasons for using a spectrum approximation sequence (such as colors indicating hot vs cold).

[GUIDE 4.19]

Use a spectrum approximation pseudocolor sequence for applications where its use is deeply embedded in the culture of users. This kind of color sequence can also be used where the most important requirement is reading map values using a key. If this sequence is used, the spacing of the colors should be carefully chosen to provide discriminable steps.

Note that if it is still important for certain data to stand out in the visualization, **you must be sure to make that detail stand out using the luminance channel**.

[GUIDE 4.20]
If it is important to see highs, lows, and other patterns at a glance, use a pseudocolor sequence that monotonically increases or decreases in luminance. If reading values from a key is also important, cycle through a variety of hues while trending upward or downward in luminance.

- **Interval Pseudocolor Sequences** - An interval sequence is one in which each unit step of the sequence represents an equal change in magnitude of the characteristic being displayed across the whole range of the sequence. A contour map (rather than a color sequence) is the traditional way of displaying an interval sequence.
- **Ratio Pseudocolors** - A ratio sequence is an interval sequence that has a true zero and all that this implies: **The sign of a value is significant.** One value can be twice as large as another. We currently do not have any technique to accurately convey ratios with any precision. However, a sequence can be designed that effectively expresses a zero point and numbers above and below zero. There are called **diverging sequences, or bipolar sequences.**
- **Bivariate Color Sequences** - Since color is three-dimensional, it is possible to display two or even three-dimensions using pseudocoloring. In such a scheme, the data values are not mapped to perceptual channels. In general, it is better to map data dimensions to perceptual color dimension. For example:

    Variable 1 → hue
    Variable 2 → saturation

  Or

    Variable 1 → hue
    Variable 2 → lightness

  Note: **bivariate color maps are notoriously difficult to read**. Generally, when we want to display two variables on the same map, it may be better to use visual texture, height difference, or another channel for one variable and color for the other, such that we map data dimensions to more perceptually separable dimensions.

<u>Application 4: Color Reproduction</u>

The problem of color reproduction is essentially one of transferring color appearances from one display device to another (computer monitor → sheet of paper). Since the human visual system is made to perceive relationships between colors rather than absolute values, the solution to the color reproduction problem lies in preserving the color relationships as much as possible, instead of the absolute values. There are a few rules that should be followed when making a color mapping from one device to another:

1) The gray axis of the image should be preserved. White on the monitor should be white on the paper.
2) Maximum luminance contrast is desirable.
3) Few colors should lie outside the destination gamut.
4) Hue and saturation shifts should be minimized.
5) An overall increase of color saturation is preferable to a decrease.

<u>October 24, 2017 (class)</u>

Note: Fine detail is always because of luminance changes, **NOT** color changes

Avoid using symbols that are too small to easily distinguish. If there are small spots, try to avoid blue and yellow as much as possible, because the background can make the symbols especially hard to see when the small size is combined with the background colors.

To account for color-blind audiences, you should avoid red-green color schemes. You should use a variation on blue-yellow.

Avoid highly saturated colors that span large areas. You should reserve high-saturation colors for small regions of data. We have a better ability to distinguish between color categories when saturation is lower.

You want to use a **perceptually ordered** color map when the color represents numerical values. If you need to know that the value for some region is higher or lower than another region's value, you want the color scheme to inherently indicate an ordering (think divergent color map). **Trumbo's univariate principle says ordered values should be represented by perceptually ordered colors, and significantly different levels should be represented by distinguishable colors.**

Make sure to **not** use a rainbow color map. Note: Rainbow Map Considered Harmful. Rainbow color map has some advantages:

- Cultural meaning (?)
- Has a lot of relatively distinct values (more colors to use as labels)
  - Helps us figure out absolute value of specific data points

However, you **don't** want to use the rainbow color map because:

- Luminance ramp makes some values appear more important(?)
  - The luminance channel for the rainbow color map is not linear at all. It's almost completely unordered, since it's not controlled in this color map.
  - Also, **you need luminance to see shape. The shape of an object is only due to the luminance. Luminance controls your perception of shape.** If you need to apply a color map to a 3D viz, do it only if the luminance is held constant (isoluminant colormap) (refer to slide with green/red 3D viz)
- Not a uniform scale - blues cover a larger area than the other colors.

- The color map can make you think there is an implied categorizing of the data based on the colors (the pyramid image in the slides)
- Not perceptually ordered
- Removes detail, since not all data variations are visually represented. This happens in part due to where the color changes happen, since the change between blue to indigo, for example, is a more noticeable change. But the change from red to orange is not as noticeable.
- Cyclic, so opposite ends are still quite similar

Small recap even though all this info is right above: Not perceptually ordered, obscures the data (divides into categories that don't necessarily exist, not uniform (hides the data), **luminance channel is not controlled (makes data appear where it doesn't exist)**)

<u>Color conventions</u>

- **US** - red is danger, green is life
- **China** - red is life, green/white is death
- **Scientific domains** - some scientific domains have color conventions!!

<u>Hue vs Luminance</u>

- **Spatial sensitivity** - Red/green and Yellow/blue yields about one third of the detail of black/white (Ask about this?). Refer to the 2 slides!!!
- **Stereoscopic Depth** - To see depth, you can't use only hue (i.e. only with color). You need luminance to properly identify depth (think depth channel in 3D modelling). You need luminance to properly see motion
- **Temporal Sensitivity** - Moving hue-change patterns seem to move slowly.
- **Form** - Shape-from-shading is **good**. Shape-from-hue is **bad.** Need luminance changes to see shape!!!
- **Category** - Hues are actually useful here and should be used.

Trumbo's Bivariate Principle

- Rows and Columns - to preserve univariate info display parameters should not obscure one another
- Diagonal - to show a positive association, displayed colors should be grouped into three perceptual classes: diagonal, above, below

<u>Hue vs Saturation</u>

- **Single color scales** - Change a single color with all other components held constant
- **Redundant color scales** - Two or more components varied together. It reinforces signals, and it combines characteristics of simple scales.

Quiz 4 material starts here.

October 30, 2017

Difference Mechanism for Fine Discrimination

One would think that humans would not be able to discriminate small-sized orientation differences, but this is clearly not the case. When given enough time, people are able to resolve far smaller differences than they can with brief exposure. The explanation for finer discriminations is **differencing mechanism**, higher-level processes that sharpen up the output from individual receptors. This mechanism is based on inhibition. This mechanism explains why the visual system is exquisitely sensitive to differences, but not to absolute values. So, for rapid target finding, it is important that targets be **distinct in orientation by 30 degrees or more and in size by a factor of two.**

Feature Maps, Channels, and Lessons for Visual Search

Different kinds of visual properties are processed separately, so they can be thought of as forming separate feature maps, roughly at the V1 level. For example, there is a map for redness, greenness, vertical orientation, horizontal orientation, etc.

So, when we are looking for something in a visualization, we define the properties that the thing we are looking for has, and our search of the visualization is directed by this set of properties. For example, if we look at an image that has different colored symbols, our search process will likely be influenced based on the colors (we will identify and process all red symbols together, same for the other colors).

[GUIDE 5.4]

Make symbols as distinct from each other as possible, in terms of both their spatial frequency components and their orientation components.

[GUIDE 5.5]

Make symbols as distinct as possible from background patterns in terms of both their spatial frequency components and their orientation components.

Preattentive Processing and Ease of Search

Preattentive processing determines what visual objects are offered up to our attention and easy to find in the next fixation.

To find out if a pattern is preattentively distinct involves measuring the response time to find a target among a set of other symbols called **distractors**. If processing is preattentive, the time taken to find the target should be equally fast no matter how many distracting non targets there are. As a rule of thumb, anything that is processed at a rate faster than 10 msec per item is considered to be preattentive.

This is important because it is often very important that we can display certain information at a glance. The features that are preattentively processed fall into the following categories:

- Line orientation
- Line strength
- Line width
- Size
- Curvature
- Spatial grouping
- Blur
- Added marks
- Numerosity (one, two, or three objects)
- Color
- Hue
    - Intensity
- Motion
    - Flicker
    - Direction of motion
- Spatial Position
    - Two-dimensional position
    - Stereoscopic depth
- Convex/concave shape from shading

There is a risk, however, in misinterpreting the findings of psychological studies and proposing a new kind of detector for every distinct shape. So, it is not a good idea to propose a new class of detector for everything that exhibits the popout effect.

It is also important to note that **not all preattentive effects are equally strong**. **In general, the strongest effects are based on color, orientation, size, contrast, and motion/blinking**. Effects such as line curvature tend to be weaker. There are also degrees of difference: large color differences have more popout than small ones. Some popout effects occur with no instruction and are difficult to miss, but other patterns labeled as preattentive require considerable attention for them to be seen.

[GUIDE 5.6]

Use strong preattentive cues before weak ones where ease of search is critical

<u>Attention and Expectations</u>

One issue with processing patterns is that if we are not looking specifically for that pattern or symbol, it can be hard to notice it. **Humans do not perceive much unless we have a**

need to find something and a vague idea of what that something looks like. In most systems, brief, unexpected events will be missed.

The question of which visual dimensions are preattentively stronger and more salient does not have a simple answer, because it always depends on the context. For example, a right-slanted line is easy to find among a group of vertical lines, but it is harder to find among a group of vertical and left-slanted lines. **Preattentive symbols become less distinct as the variety of distractors increases**. **There are, in fact, two factors that determine whether something stands out preattentively:**

1) **The degree of difference of the target from the non targets and**
2) **The degree of difference of the nontargets from each other**

[GUIDE 5.7]

For maximum popout, a symbol should be the only object in a display that is distinctive on a particular feature channel. For example, it might be the only item that is colored in a display where everything else is black and white.

Highlight and Asymmetries

There is another issue with making targets distinctive, and it comes from **asymmetries in some preattentive factors**. For example, **adding marks to highlight a symbol is generally better than taking them away. If all the symbols in a set except for a target object have an added mark, the target will be less distinctive.** It is better to highlight one word by underlining it, rather than underlining everything except the target word.

[GUIDE 5.8]

Use positively asymmetric preattentive cues for highlighting.

When a visual design is complex (the visualization employs color, texture, and shape), the highlighting problem becomes more difficult.

[GUIDE 5.9]

For highlighting, use whatever feature dimension is used least in other parts of the design.

Motion is a useful highlighting feature that has been made much easier to implement with the advent of modern technology, but it is not always a good thing to use. In some cases, the addition of motion can be too distracting and just unnecessary.

Scroll down to see next guide

[GUIDE 5.10]

When color and shape channels are already fully utilized, consider using motion or blink highlighting. Make the motion or blinking as subtle as possible, consistent with rapid visual search.

Another way of highlighting information is the use of blur. Blurring out everything *but* the target data can make it stand out. This technique is called **semantic depth of field**, since it applies the depth-of-focus effects that can be found in photography to the display of data according to the semantic content. However, it is important to note that by blurring, it is possible to make important information ineligible.

Coding with Combinations of Features

Often, we wish to make objects distinctive using two or more channels. There are two issues with this, though. The first issue is with using redundant coding for extra distinctiveness. The second issue is that we don't know what we can expect if we use more complex patterns in symbol design.

Coding with Redundant Properties

Making something more distinct on multiple feature dimensions (such as color, shape, and size all at once) is known as redundant coding. It means that someone can search based on any or all of the properties. Sometimes the degree to which the search is improved is not entirely worth it. That being said, there is almost always a benefit to redundant coding.

[GUIDE 5.11]

To make symbols in a set maximally distinctive, use redundant coding wherever possible. For example, make symbols differ in both shape and color.

What Is Not Easily Findable: Conjunctions of Features

We have only discussed what features are easily distinguishable, so far. We have not mentioned what features are hard to distinguish until now. **A conjunction search is one in which the user must search for the specific conjunction of two features (such as searching for a symbol that is *both* red and square).** This is very different from redundant coding, where parallel search can be carried out on one **or** the other feature. In general, conjunction searches are slow and not preattentive. This means we cannot learn to rapidly find more complex patterns regardless of how much experience we have doing that.

[GUIDE 5.12]

If symbols are to be preattentively distinct, avoid coding that uses conjunctions of basic graphical properties.

Highlighting Two Data Dimensions: Conjunctions That Can Be Seen

It is important to note that **not all conjunction searches are not preattentive. There are some combinations of features that are still preattentive**. Some preattentive conjunction searches are the following:

- **Spatial grouping on the XY plane** - Preattentive search can be guided by the identification of spatial clusters. Thus, conjunction of space and color can be search preattentively. See the image on page 160
- **Stereoscopic depth** - Conjunction of stereoscopic depth and color, or of stereoscopic depth and movement, can be preattentively processed.
- **Luminance polarity and shape** - Luminance polarity with targets lighter and darker than gray background can support a preattentive conjunction search. See the figure on page 161.
- **Convexity, concavity, and color** - Conjunction of perceived convexity and color can be preattentively processed. In this case, the convexity is perceived through shape-from-shading information.
- **Motion** - Motion and target shape can be preattentively scanned conjunctively. Thus, if the whole set of targets is moving, we don't need to look for nonmoving targets. We can preattentively find, for example, the red moving target.

One application in which preattentive spatial conjunction may be useful is found in geographic information systems (GISs). In these systems, data is often characterized as a set of layers. For example, a layer representing the surface topography, a layer representing minerals, and a layer representing ownership patterns. Such layers may be differentiated by means of motion or stereoscopic depth cues.

[GUIDE 5.13]

When it is important to highlight two distinct attributes of a set of entities, consider coding one using motion or spatial grouping and the other using a property such as color or shape.

October 31, 2017 (class)

Preattentive Processing

Preattentive processing is what characteristics make a particular shape appear in a pattern of other shapes that are flashed briefly in front of your eyes.

- **Orientation** - Changing the orientation of symbols is a good way to make them distinct. **Targets should be >= 30 degrees to be visually distinct.**
- **Size** - Likewise, changing the size of symbols makes them easier to identify. **Targets need to be >= 2x the size to be visually distinct.**

Differences in spatial frequency allows us to differentiate between symbols more easily (think of the Xs and circles scatter plot).

<u>Visual Channels</u>

- **Orientation and size** - These channels jump out and fire in your visual cortex the most (with luminance)
- **Color** - There are neurons specifically for colors, processed via the opponent processing channel mechanisms. The neurons fire for each color, and will make certain colors jump out if they are surrounded by colors that fire the opposite neurons (green/red combo or yellow/blue combo). It triggers the spike in differences
- **Elements of local stereoscopic depth** - How far away or close something is is determined by neurons that fire for depth cues. These neurons just don't work for some people
- **Elements of local motion** - Motion is also detected in our visual channels

Use different visual channels to display aspects of data so that they are visually distinct.

To make symbols easy to find, make them distinct from their background and from other symbols. For example, the primary spatial frequency of a symbol should be different from the spatial frequency of the background texture and from other symbols. This refers to the static background image in the slides.

Make symbols as distinct from each other as possible, in terms of both their spatial frequency components and their orientation components.

Make symbols as distinct as possible from their background patterns in terms of both their spatial frequency components and their orientation components.

The following are all channels that aid in preattentive processing. You want to refer to this list (rather, the one above from book notes that actually includes all of them) when you need to distinguish your symbols in a visualization:

- Orientation
- Curved/straight
- Shape
- Size
- Color
- Light/dark
- Enclosure

- Convex/concave
- Adition

The following are **not** preattentively processed:
- Juncture (joining of lines)
- Parallelism (of lines)

Use strong preattentive cues before weak ones where ease of search is critical. Strong cues: Color, orientation, size, contrast, and motion/blinking.

Asymmetric Preattentive Cues
Think back to the underlining of a whole sentence vs just underlining one word.

It is better to add than to take away. Add markers to points of interest to make them distinct.

For highlighting, use whatever feature dimension you used least in other parts of the design.

When color and shape channels are already fully utilized, consider using motion or blink highlighting. Make the motion or blinking as subtle as possible, consistent with rapid visual search. If something is blinking/moving too fast, it makes it too difficult to focus on other parts of the visualization.

To make symbols in a set maximally different, use redundant coding wherever possible; for example, make symbols differ in both shape and color.

If symbols are to be preattentively distinct, avoid coding that uses conjunctions of basic graphical properties.

When it is important to highlight two distinct attributes of a set of entities, consider coding one using motion or spatial grouping and the other using a property such as color.

Luminance is very good at highlighting stuff.

November 6, 2017
Integral and Separable Dimensions: Glyph Design
Sometimes we need a symbol to do more than simply stand for something. Sometimes we want the symbol to convey an aspect of the data (temperature, size, etc). **When a symbol**

represents quantity, it is known as a glyph. To create a glyph, one or more quantitative data attributes are mapped in a systematic way to the different graphical properties of an object.

The concept of integral vs. separable visual dimensions tells us when one display attribute (such as color) will be perceived independently from another (such as size). With **integral display dimensions, two or more attributes of a visual object are perceived holistically and not independently.** One example is a rectangle, which is perceived as a holistic combination of the rectangle's height and width. Another example is yellow light, which is the combination of red and green (And we have trouble distinguishing between the red and green parts of the yellow light). **With separable dimensions, people tend to make separate judgements about each graphical dimension. This is sometimes called analytic processing.** One example of this is the diameter and color of a circle.

[GUIDE 5.14]

If it is important for people to respond holistically to a combination of two variables in a set of glyphs, map the variables to integral glyph properties.

[GUIDE 5.15]

If it is important for people to respond analytically to a combination of variables, making separate judgements on the basis of one variable or the other, map the variables to integral glyph properties.

Representing Quantity

Visual quantities that increase continuously are said to be **monotonic**. Some visual qualities are **not** monotonic, such as orientation (you can't say one orientation is more or less than another). Monotonic display variables naturally express relations such as great than or less than if they have a quality that we associate with increasing value.

[GUIDE 5.16]

When designing a set of glyphs to represent quantity, mapping to any of the following glyph attributes will be effective: size, lightness (on a dark background), darkness (on a light background), vividness (higher saturation) of color, or vertical position in the display

When a data variable is mapped to some visual attribute such as length, area, volume, or lightness, we can judge relative quantities at a glance, but we cannot do so very accurately.

[GUIDE 5.17]

Ideally, use glyph length or height, or vertical position, to represent quantity. If the range of values is large, consider using glyph area as an alternative. Never use the volume of a three-dimensional glyph to represent quantity.

Representing Absolute Quantities

Since we are generally interested in seeing patterns when we make a visualization, it is generally more useful for us to be able to see relative differences in the data points, rather than knowing the absolute values of the data points. If we do need to know the actual values of things, there are a few solutions we can employ. We can simply add numbers to a glyph, or use a numerical scale. Unless it is done carefully, however, the numbers will add visual noise and may obscure important patterns in the data. Another solution is to create a glyph that by its shape conveys numerical values. One good example of this is the wind barb.

Multidimensional Discrete Data: Uniform Representation versus Multiple Channels

[GUIDE 5.18]

In general, the use of heterogeneous display channels is best combined with meaningful mappings between data attributes and graphical features of a set of glyphs

Useful Field of View

When we are scanning an object or area for information, the size of the region from which we can quickly take in information changes size according to the size of the area we are scanning. This is known as the **useful field of view (UFOV).** The UFOV varies depending on the task at hand and the information being displayed. The UFOV roughly varies with target density to maintain a constant number of targets in the attended region. With greater target density, the UFOV becomes smaller and attention is more narrowly focused. With a low target density, a larger area can be attended.

[GUIDE 5.19]

When designing user interrupts, peripheral altering cues must be made strong if the cognitive load is expected to be high.

The Role of Motion in Attracting Attention

The UFOV has been suggested to be far larger for detection of moving targets than for detection of static targets. As time goes on, we find an increasing need for signals that can attract a user's attention on the screen. In this case, we utilize **user interrupts to alert the user of additional or more important information.** There are four basic visual requirements for a user interrupt:

1) A signal should be easily perceived, even if it is outside of the area of immediate focal attention
2) If the user wishes to ignore the signal and attend to another task, the signal should continue to act as a reminder
3) The signal should not be so irritating that it makes the computer unpleasant to use
4) It should be possible to endow the signal with various levels of urgency

Essentially, the problem is figuring out how to attract a user's attention to information outside of the central parafoveal region of vision. There are multiple factors that limit the ways in which we can implement user interrupts. We have a low ability to detect small targets in the periphery of the visual field. Additionally, peripheral vision is color blind. Furthermore, saccadic suppression (blindness during eye movements) means that some transitory event occurring in the peripheral will generally be missed if it occurs during a saccadic movement. Based on these facts, it seems that a single, abrupt change in the appearance of an icon is unlikely to be an effective signal.

There are two possible solutions when the above hindrances are considered. The first solution is to use auditory cues. This is useful in some cases, but are not in the scope of this course. The other solution is to use blinking or moving icons. One downside of this is that users seem to find flashing/blinking/ notifications annoying. In this case, moving targets may be a better solution. The important thing to take away from the effectiveness of motion is **that the appearance of a new object is the reason why a user interrupt is effective.** To achieve the most effective user interrupt, it is probably best to use some notification that moves into view, disappears, and then reappears every so often.

November 7, 2017 (class)

Hexadecimal stuff and memory

Displays

Your monitor has pixels (wow!). Each pixel requires one bit of memory (on or off). With this one bit, you can display 2 colors (black and white for on and off). For a 10x10 display, you need 100 bits of memory.

With 3 color displays, you have 3 pixels and can now have 8 colors. In a 10x10 display, you need 300 bits of memory.

There's some stuff the slides. Look at it. This is 100% gonna be on the next quiz.

Data Dimensions

Some data dimensions are as follows:

- Color
- Size
- Orientation

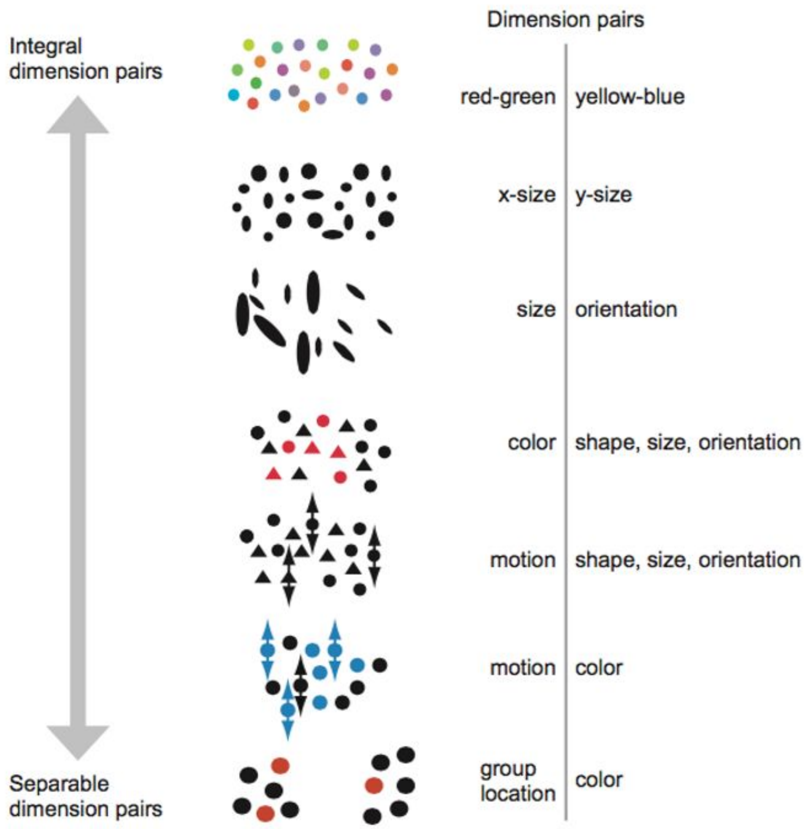
- Shape
- Motion
- Group

The big question is which dimensions can we combine or not combine? **Are they integral or separable?** Questions to ask when determining this: Will the color coding scheme interfere with our perception of glyph size? Will using both color and size make the information clearer?

- **Integral Data Dimensions -** Two or more attributes of a visual object are perceived holistically and not independently. I.e. you cannot separate these two things in your mind. For example, you can't tell how much red and green compose a certain hue of yellow. Some integral data dimensions are: **color, shape (width and height). Note: not all colors are integral. Red and blue are separable (you can add them together). Two colors directly across from each other in the color wheel are integral.**
- **Separable Data Dimensions -** People tend to make separate judgements about each data dimension. Some separable data dimensions are: **groups, motion and color combined (motion and a lot of things, actually), color and size (height and width).**

This matters because you need to be careful not to represent different data using integral dimensions if the data is not actually related. **If it is important for people to respond holistically to a combination of two variables in a set of glyphs, map the variables to integral glyph properties (you get to make the decision as to whether or not it's important for the user to respond holistically or not). If it is important for people to respond analytically to a combination of variables, making separate judgments on the basis of one variable other other, map the variables to integral glyph properties.**

SCROLL DOWN



Note: Don't use three-dimensional visualizations when the display medium is two-dimensional

- **Ideally use length or height, or vertical position, to represent quantity**
- **If range of values is large, consider using glyph areas as an alternative**
- **Never use the volume of a three-dimensional glyph to represent quantity**

November 8, 2017

Static and Moving Pattern

Gestalt Laws

This is a set of clear descriptions of many basic perceptual phenomena. They are laws about pattern perception. The Gestalt laws easily translate into a set of design principles for visualizations. We will discuss 8 of these laws:

1) **Proximity -** Spatial proximity is a powerful perceptual organizing principle, and is one of the most useful in design. Things that are close together are perceptually grouped together. This is known as the **spatial concentration principle: we perceptually group regions of similar element density.**

[GUIDE 6.1]

Place symbols and glyphs representing related information close together.

There is also a **perceptual efficiency** to using proximity. We more readily pick up information close to the fovea, so less time and effort will be spent in neural processing and eye movements if related information is spatially grouped.

2) **Similarity -** The shapes of individual pattern elements can also determine how they are grouped. **Similar elements tend to be grouped together.**

[GUIDE 6.2]
When designing a grid layout of a data set, consider coding rows and/or columns using low-level visual channel properties, such as color and texture.

3) **Connectedness -** Connectedness can be a more powerful grouping principle than proximity, color, size, or shape. **Connecting different graphical objects by lines is a very powerful way of expressing that there is some relationship between them**.

[GUIDE 6.3]
To show relationships between entities, consider linking graphical representations of data objects using lines or ribbons of color.

4) **Continuity** - The Gestalt principle of continuity states **we are more likely to construct visual entities out of visual elements that are smooth and continuous, rather than ones that contain abrupt changes in direction**. This principle can be applied to visualizations in which you need to draw networks of nodes and links between them. It is easier to identify the source and target of links if they are smooth
5) **Symmetry** - Symmetry can be powerful for organizational purposes. To take advantage of symmetry, the important patterns must be small. We are most sensitive to symmetrical patterns that are small, less than 1 degree in width and 2 degrees in height, and centered around the fovea. Note that we more rapidly perceive symmetry around vertical and horizontal axes.

[GUIDE 6.4]
Consider using symmetry to make pattern comparisons easier, but be sure that the patterns to be compared are small in terms of visual angle (< 1 degree horizontally and < 2 degrees vertically). Symmetrical relations should be arranged on horizontal or vertical axes unless some framing pattern is used.

6) **Closure and Common Region** - A closed contour tends to be seen as an object. We have a tendency to close contours that have gaps in them. When there is a perceived closed

contour, there is a strong tendency to divide the regions of space into “inside” and “outside”. **A region enclosed by a contour becomes a common region**. When the boundary of a contour-defined region becomes complex, what is inside or outside may become unclear. In these cases, using color, texture, or Cornsweet contours will be more effective.

[GUIDE 6.5]
Consider putting related information inside a closed contour. A line is adequate for regions having a simple shape. Color or texture can be used to define regions that have more complex shapes.

[GUIDE 6.6]
To define multiple overlapping regions, consider using a combination of line contour, color, texture, and Cornsweet contours.

7) **Figure and Ground - A figure is something objectlike that is perceived as being in the foreground. The ground is whatever lies behind the figure.** In general, smaller components of a pattern tend to be perceived as objects. This kind of perception is part of the fundamental perceptual act of identifying objects.

[GUIDE 6.7]
Use a combination of closure, common region, and layout to ensure that data entities are represented by graphical patterns that will be perceived as figures, not ground.

Contours Continued

The brain is exquisitely sensitive to the presence of contours. A contour can be defined by a line, a boundary between regions of different colors, stereoscopic depth, motion patterns, or the edge of a region of a particular texture. The theory underlying contour perception is that there is mutual reinforcement between neurons that have receptive fields that are smoothly aligned; there is inhibition between neurons with non aligned receptive fields. The result is a kind of winner-take-all effect. Stronger contours beat out weaker contours.

[GUIDE 6.8]
For vector field visualizations, use contours tangential to streamlines to reveal the orientation component

[GUIDE 6.9]
To represent flow direction in a vector field visualization, use streamlets with heads that are more distinct than tails, based on luminance contrast. A streamlet is a glyph that is

elongated along a streamline and which induces a strong response in neurons sensitive to orientations tangential to the flow.

[GUIDE 6.10]
For vector field visualizations, use more distinct graphical elements to show greater field strength or speed. They can be wider, longer, more contrasting, or faster moving.

November 9, 2017 (Class)

Gestalt Laws

These are robust rules that describe the way we see patterns. Gestalt means pattern in German. As usual, read the slides because they have the images from the book that I can't get an electronic version of elsewhere.

- **Proximity** - Place symbols and glyphs representing related information close together
- **Similarity** - When designing a rigid layout of a data set, consider coding rows and/or columns using low-level visual channel properties such as color and texture
- **Connectedness** - To show relationships between entities, consider linking graphical representations of data objects using lines or ribbons of color. **Lines are really good at showing the connection between data objects (like, better than everything else pretty much).**
- **Continuity** - We are more likely to construct visual entities out of the visual elements that are smooth and continuous, rather than ones that contain abrupt changes in direction.
- **Symmetry** - Consider using symmetry to make pattern comparisons easier, but be sure that the patterns to be compared are small in terms of visual angle (< 1 degree horizontally and < 2 degrees vertically). Symmetrical relations should be arranged on horizontal or vertical axes unless some framing pattern is used.
- **Closure** - Consider putting related information inside a closed contour. A line is adequate for regions having a simple shape. Color or texture can be used to define regions that have more complex shapes. To define multiple overlapping regions, consider using a combination of line contour, color, texture, and Cornsweet contours. Be careful with only using colors to define groups, because if they overlap it can become too hard to distinguish that an area is an overlap rather than its own region.
- **Figure and Ground** - Use a combination of closure, common region, and layout to ensure that data entities are represented by graphical patterns that will be perceived as figures, not ground. Distinctions between the background and foreground are usually defined by the area of the regions. Having roughly equal areas of 2 regions allows us to easily switch between which region is the foreground and which is the background.