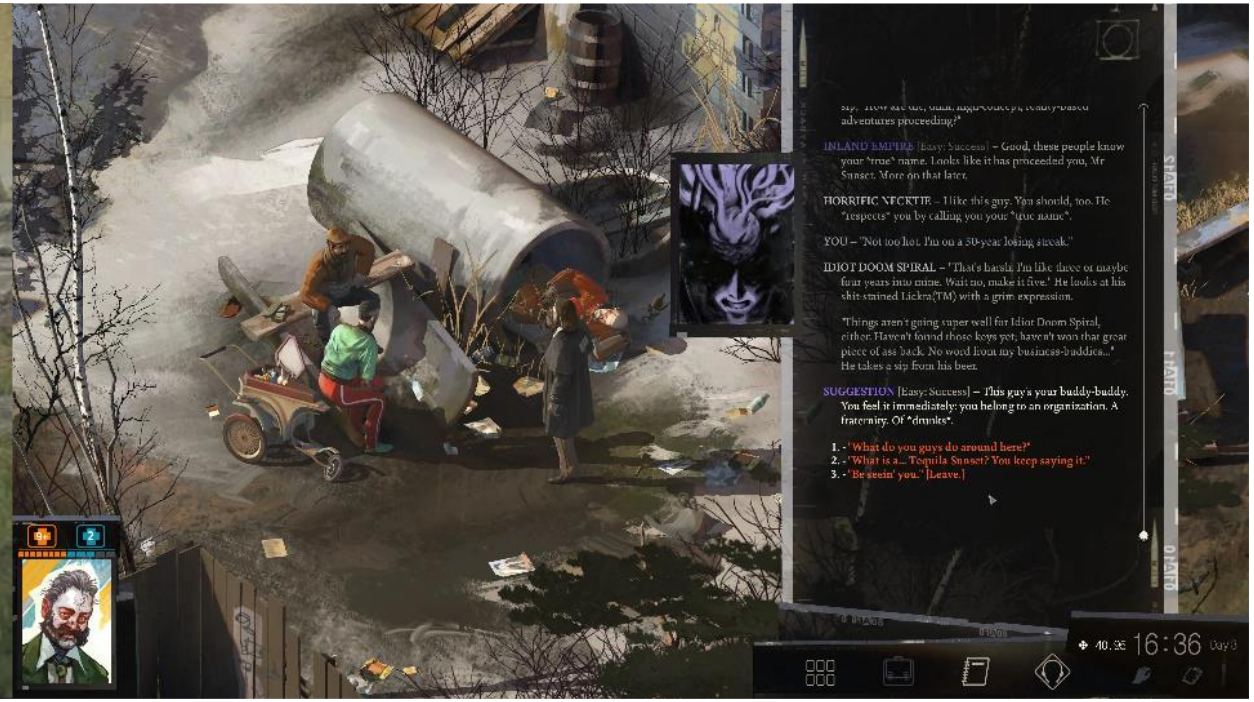
8. Font and Game

Game Engineering & XR Technology
Prof. HyeongYeop Kang
siamiz@khu.ac.kr
IIIXR LAB

Introduction

Font and Font Rendering

- Conveying text information is one of the most important components of games.
- Providing a tutorial, guide, or scenario is all achieved by conveying text information to users.
- In digital devices such as computers and smartphones, fonts are exploited to print text onto the display in a specific style and size.
- In this lecture, we explore how the fonts are defined in computers and how they are processed to be visualized onto the displays.

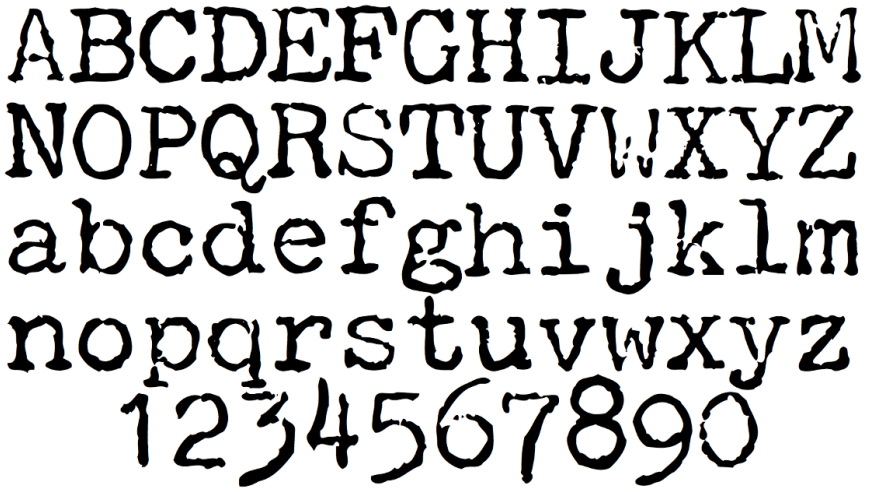


Font and Typeface

What is a Font?

- The word font refers to a set of printable or displayable typography or text characters in a specific style and size.

- Typography is the art and technique of arranging type to make written language legible, readable, and appealing when displayed.



What is a Typeface?

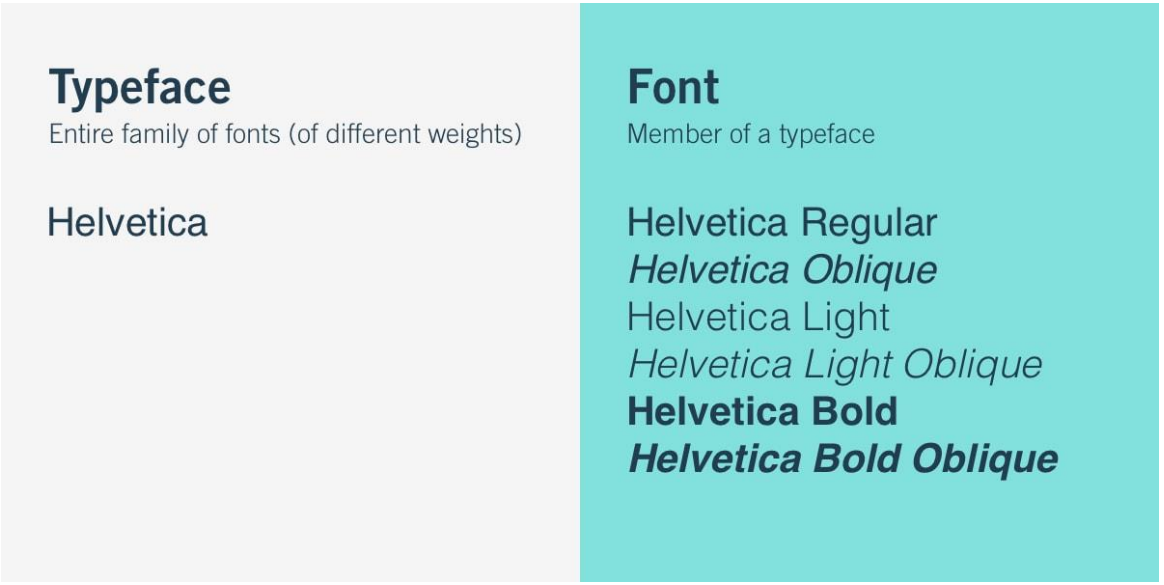
- Typeface refers to a collection of characters, letters and numbers that share the same design.
  - While each letter is unique, certain shapes are shared across letters.
  - A typeface represents shared patterns across a collection of letters.

Font and Typeface

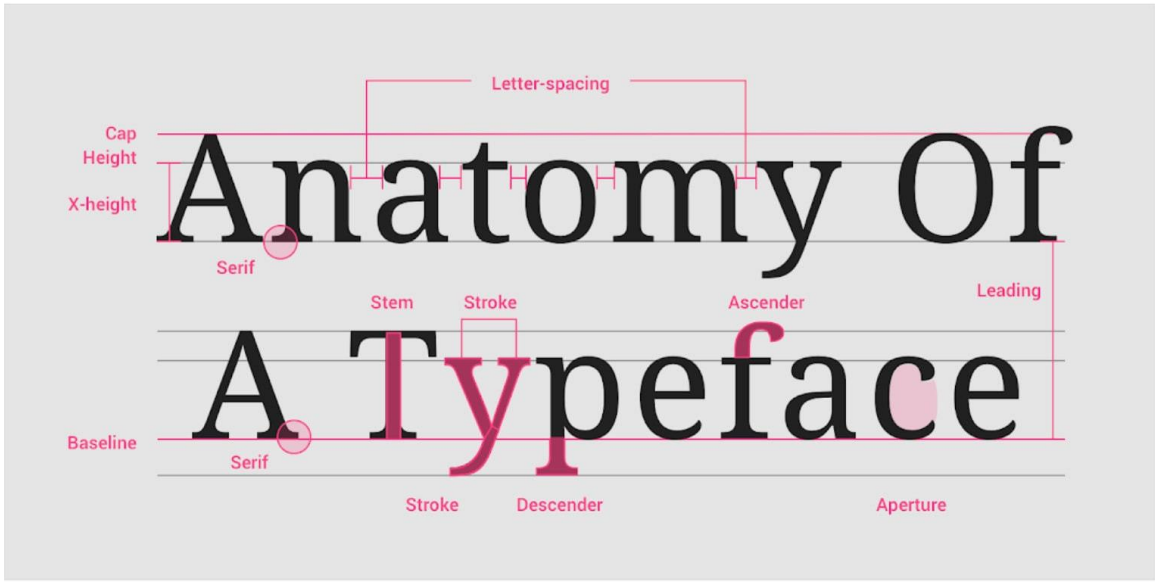
Font vs. Typeface

- In professional typography, the term typeface is not interchangeable with the world font, because the term font has historically been defined as a given alphabet and its associated characters in a single size[1].
  - For example, 8-point Caslon Italic was one font, and 10-point Caslon Italic was another.

- Instead, typeface only describes a particular style of lettering. It describes the word with design features that characterize a particular style of lettering.



font vs. typeface



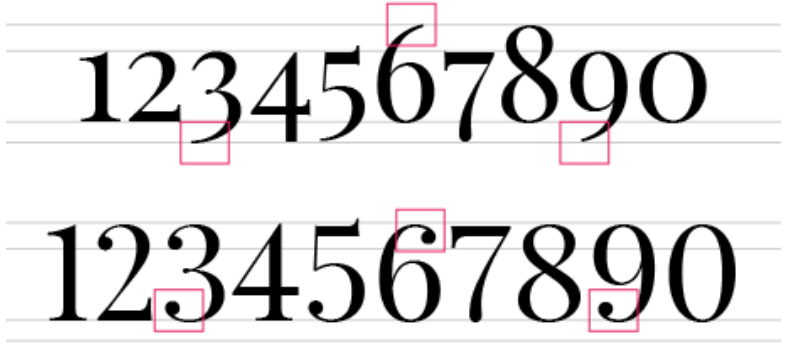
design features

Typeface Design Features

Baseline[2][3]

- In European and West Asian typography, the baseline is the line upon which most letters sit and below which descenders extend.
  - In the example to the right, the letter ‘p’ has a descender; the other letters sit on the baseline.

- Most typefaces are similar in the following ways as regards the baseline:
  - Capital letters sit on the baseline. The most common exceptions are the J and Q.
  - Lining figures (such as Arabic numerals) sit on the baseline.
  - The following text figures have descenders: 3 4 5 7 9.
  - The following lowercase letters have descenders: g j p q y.
  - Glyphs with rounded lower and upper extents (0 3 6 8 c C G J o O Q) dip very slightly below the baseline ("overshoot") to create the optical illusion that they sit on the baseline, and rise above the x-height or capital height to create the illusion that they have the same height as flat glyphs (such as those for H x X 1 5 7). Peter Karow's Digital Typefaces suggests that typical overshoot is about 1.5%.



- A text’s baseline must sit on the 4dp grid. Line-height must be a value divisible by 4 to maintain the grid.

Typeface Design Features

Body Height, Cap Height, and X-height[4][5]

- The body height refers to the distance between the top of the tallest letterform to the bottom of the lowest one.

- The cap height is the height of a typeface's flat capital letter (such as M or I) above the baseline.
  - Round and pointed capital letters, such as S and A, are optically adjusted by being drawn with a slight overshoot above the cap height to achieve the effect of being the same size.
  - Every typeface has a unique cap height.

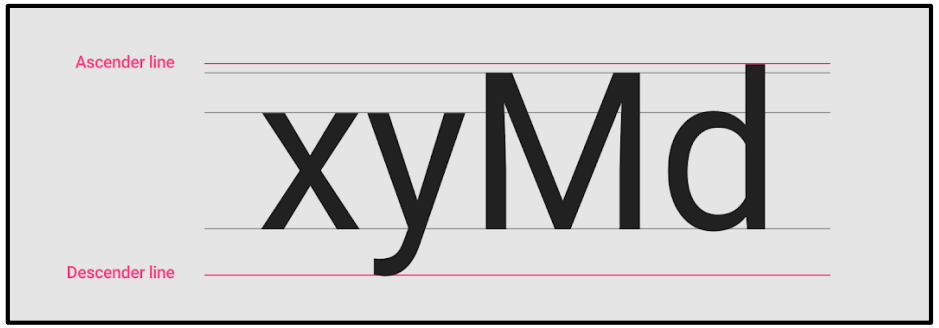
- The height of the small letters is the x-height. It indicates how tall or short each glyph in a typeface will be.
  - Typefaces with tall x-heights have better legibility at small font sizes, as the white space within each letter is more legible.

Typeface Design Features

Ascenders and descenders

- Ascenders and descenders increase the recognizability of words.
  - For this reason, many situations that require high legibility such as road signs avoid using solely capital letters (i.e. all-caps).
  - Studies made at the start of the construction of the British motorway network concluded it and the use of mixed-case letters became universal across the UK.

- Ascenders are an upward vertical stroke found in certain lowercase letters that extend beyond either the cap height or baseline.



- Descenders is the portion of a letter that extends below the baseline of a font.
  - In most fonts, descenders are reserved for lowercase characters such as g, j, q, p, y, and sometimes f.
  - Some fonts also use descenders for some numerals (typically 3, 4, 5, 7, and 9). Such numerals are called old-style numerals.
  - Some fonts also use descenders for the tails on a few uppercase letters such as J and Q.

descenders

Typeface Design Features

Weight

- Weight refers to the relative thickness of a font's stroke.

- A typeface can come in many weights; and four to six weights is a typical number available for a typeface.
  - The figure on the right shows four types of common weights: 1) light, 2) regular, 3) medium, and 4) bold.

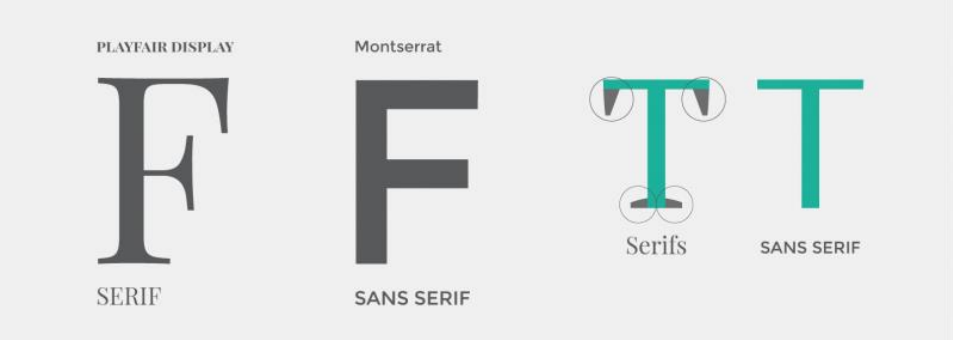
Typeface Serifs

Serifs

- Typeface can be divided into two main categories: serif and sans serif.
  - Serifs comprise the small features at the end of strokes within letters.
  - Sans serif have no serifs (from French *sans*, meaning without).

- Typefaces with serifs are often considered easier to read in long passages than those without.
  - As a general rule, printed works such as newspapers and books almost always use serif typefaces, at least for the text body.
    - ✓ Some popular examples of serif typefaces are Times New Roman, Garamond, and Georgia.
  - Web sites (that do specify a font) commonly use modern sans serif fonts, because it is commonly believed that sans serif fonts are easier fonts to read on the low-resolution computer screen.
    - ✓ Some popular sans-serif fonts are Arial, Futura, and Helvetica.

- Typefaces with serifs are sometimes considered more classic or formal, and sans-serif fonts are often considered more minimalist or casual.

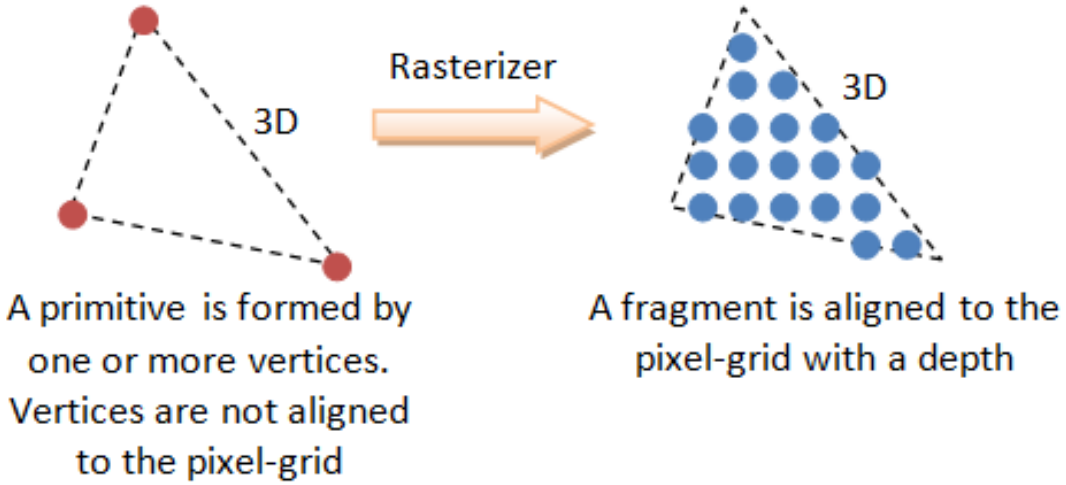


Font Rasterization

Rasterization

- Rasterization is the task of taking an image described in a vector graphics format (shapes) and converting it into a raster image (a series of pixels, dots or line).

- A common representation of digital 3D models is polygonal.
  - Before rasterization, individual polygons are broken down into triangles, therefore a typical problem to solve in 3D rasterization is rasterization of a triangle.



Font Rasterization

Font rasterization

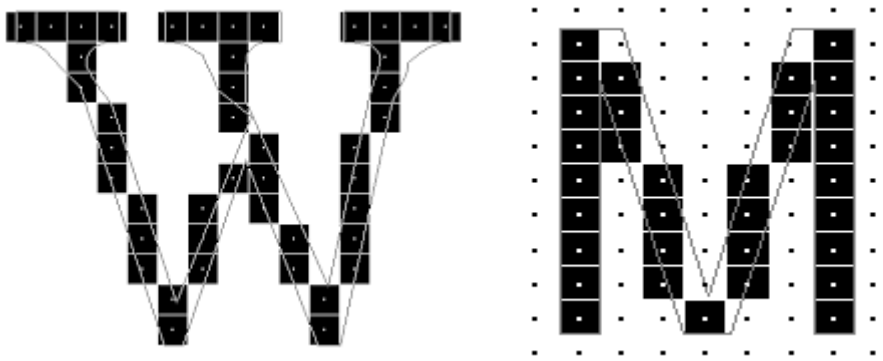
- Font rasterization is the process of converting text from a vector description to a raster or bitmap description.

- This often involves some anti-aliasing on screen text to make it smoother and easier to read.

Bi-level rendering (rasterization)

- The simplest form of rasterization is simple line-drawing with no anti-aliasing.

- In Microsoft's terminology, this is called bi-level rendering (also called black/white rendering).
  - This form of rendering is also called aliased or jagged.
  - This is the fastest rendering method.
  - However, the rendered glyphs may become hard to recognize at small sizes.



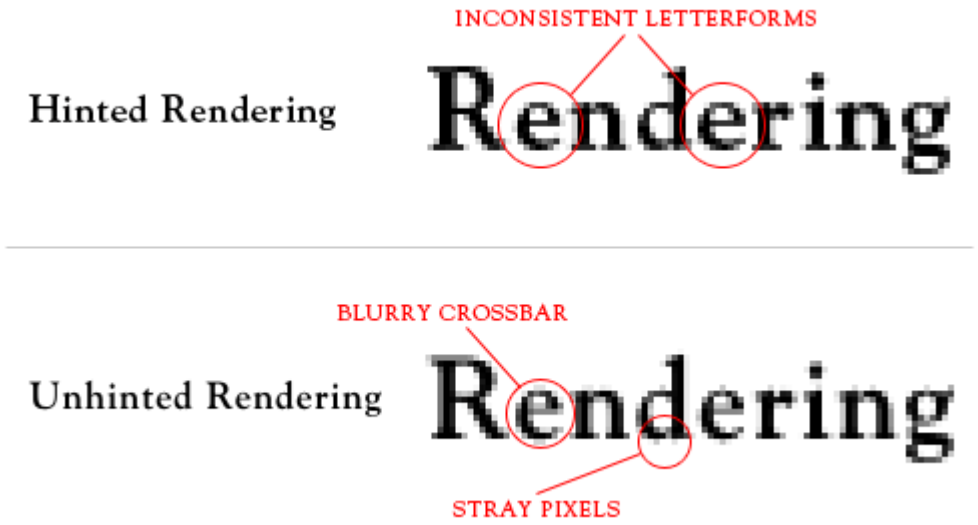
bi-level rendering

Font Rasterization

Font hinting

- To improve readability, many font data files (such as TrueType) contain *hints* that help the rasterizer decide where to render pixels for particularly troublesome areas in the glyphs.

- Hinting is the programming instructions that fine-tune a font's rasterization.
  - Hinting can control the heights and widths of a font's uppercase and lowercase letters, the widths of its individual lines, the amount of white space around letters, etc.

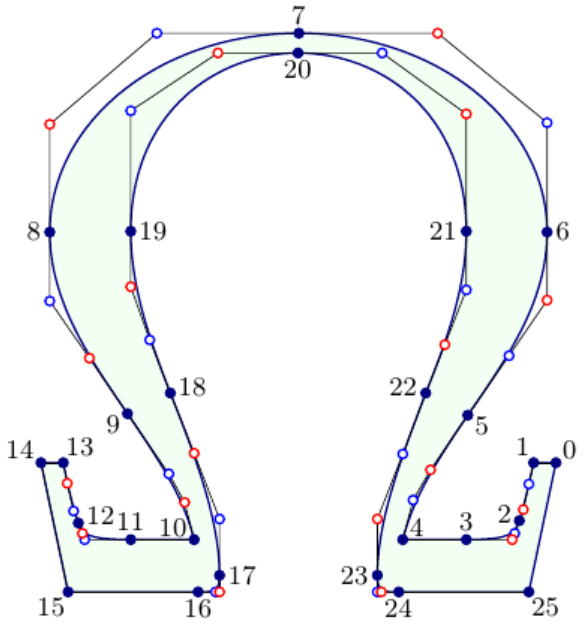


Font Rasterization

PostScript fonts

- Even when fonts are hinted optimum, onscreen results are still not guaranteed, since different font technologies approach hinting differently.

- In the early 1980s, Adobe developed a scalable font technology called PostScript fonts.
  - In PostScript system, the glyphs are described with cubic Bezier curves, and thus a single set of glyphs can be resized through simple mathematical transformations.
  - Simply to say, font scaling strategy is handled not by the fonts, but by the rasterizer.
  - Fonts in PostScript format look often good with relatively simple hinting or no hinting at all.

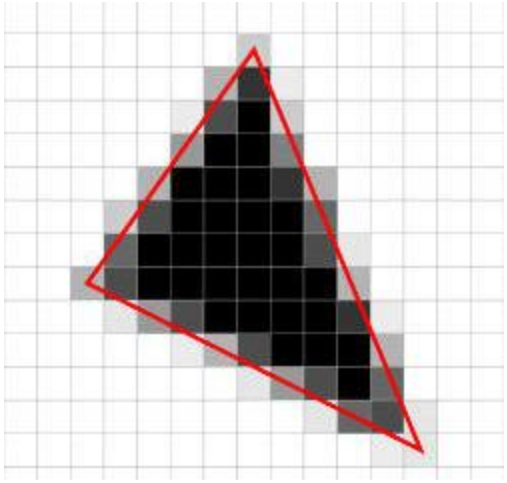


Font Rasterization

Anti-aliasing and font rendering

- A more complicated approach is to use anti-aliasing techniques from computer graphics.

- This can be thought of as determining, for each pixel at the edges of the character, how much of that pixel the character occupies, and drawing that pixel with that degree of opacity.
  - When drawing a black (000000) letter on a white (FFFFFF) background, if a pixel ideally should be half filled, it is drawn 50% gray (BCBCBC).
  - If the letter includes a vertical line that should be one pixel wide but falls exactly between two pixels, it appears on screen as a two-pixel-wide gray line.



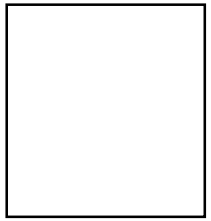
half filled pixel is drawn 50% gray

Font Rasterization

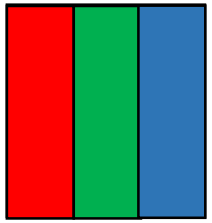
Subpixel font rendering

- Another promising strategy of anti-aliasing is to use subpixel rendering.

- Most computer displays have pixels made up of multiple subpixels (typically one each for red, green, and blue, which are combined to produce the full range of colors).
  - For example, a horizontal resolution of 512 pixels is actually composed of 512 red, 512 green, and 512 blue sub-pixels.
  - The human eye doesn't see closely spaced colors individually since our vision system mixes these three primary colors.
  - Sub-pixel rendering exploit this characteristic by rendering at the subpixel resolution rather than using whole pixels, which can increase the effective resolution of the screen.



our eye sees
a single pixel

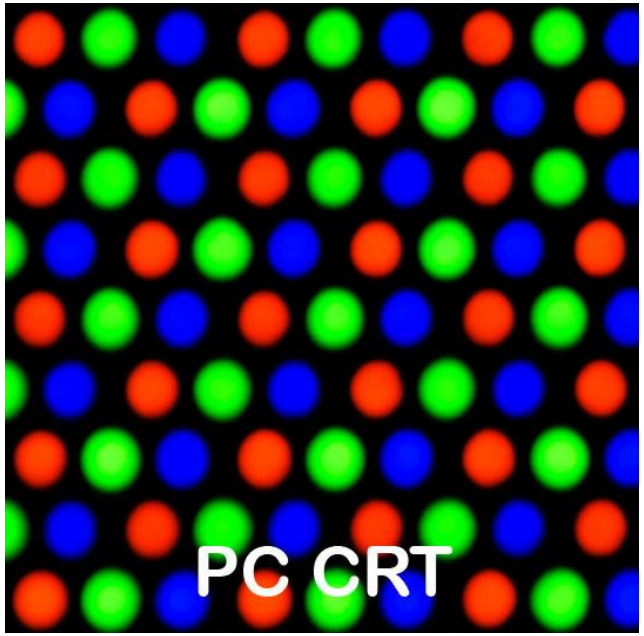


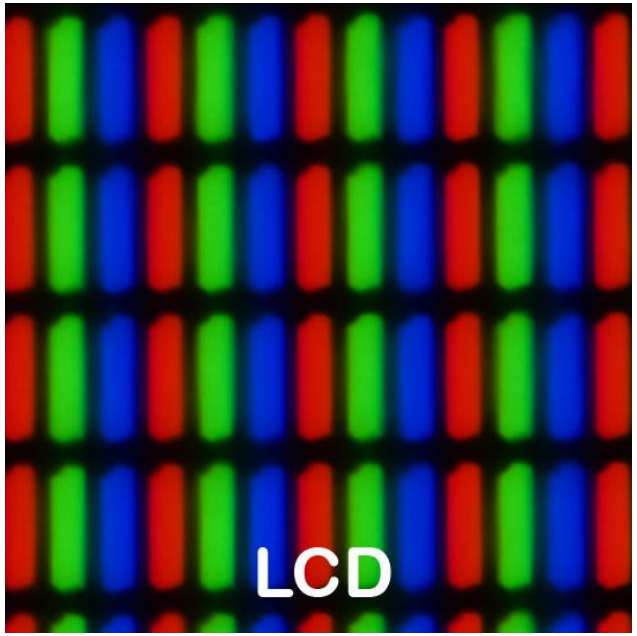
actual pixel when
magnifying glass

Font Rasterization

Subpixel font rendering

- Subpixel rendering is better suited to some display technologies than others.
  - The technology is well-suited to LCDs and other technologies where each logical pixel corresponds directly to three or more independent colored subpixels, but less so for CRTs.
  - In a CRT the light from the pixel components often spreads across pixels, and the outputs of adjacent pixels are not perfectly independent.
  - If a designer knew precisely about the display's electron beams and aperture grill, subpixel rendering might have some advantage but the properties of the CRT components, coupled with the alignment variations that are part of the production process, make subpixel rendering less effective for these displays.

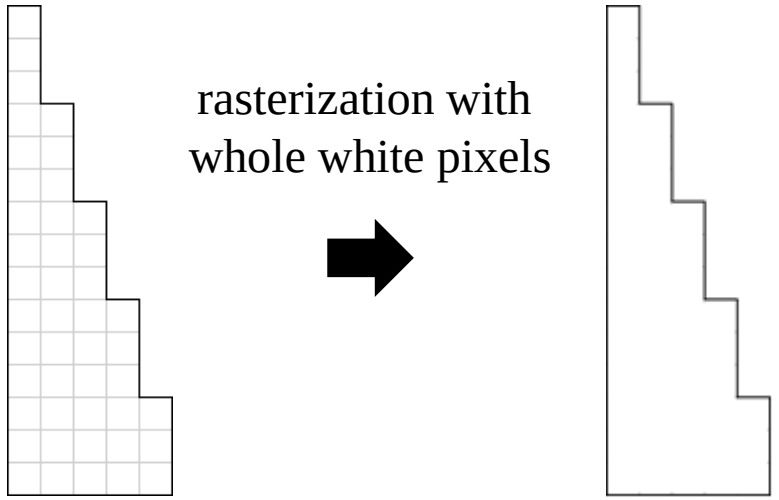




Font Rasterization

Subpixel font rendering

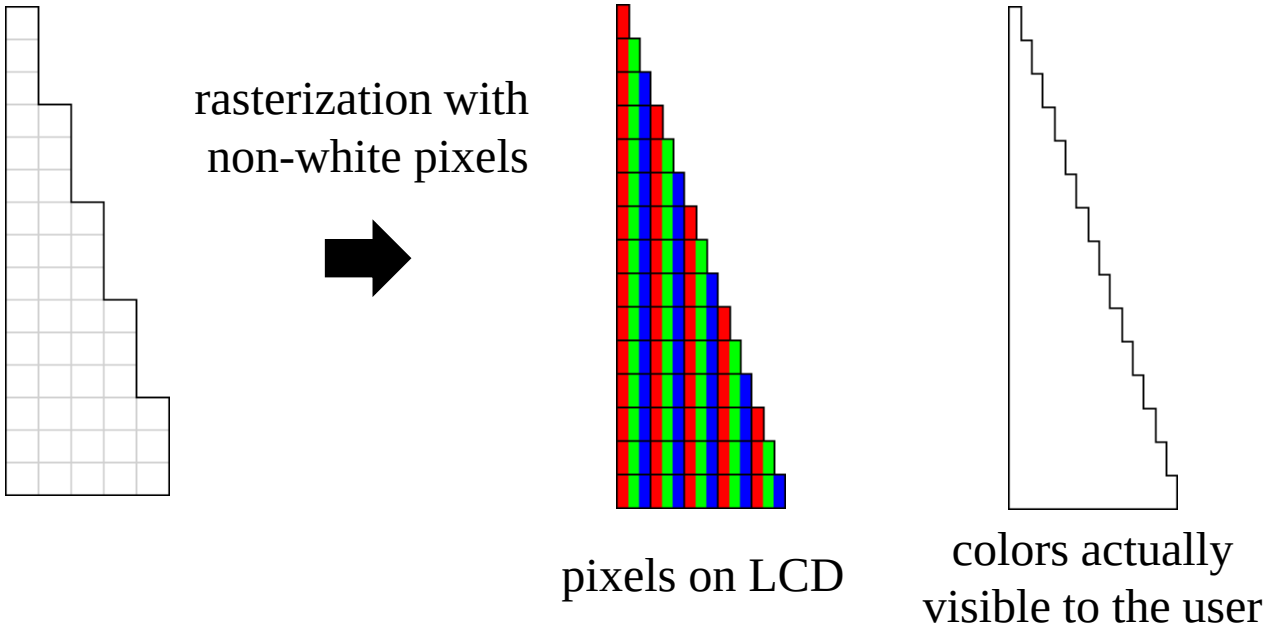
- Here is a good example[7]:
  - Suppose we wish to draw an object with a sloping diagonal edge. Using standard ‘whole’ white pixels, the best job we can do creates a very jagged edge.



Font Rasterization

Subpixel font rendering

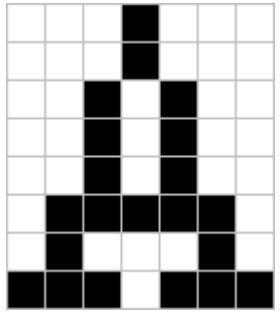
- Here is a good example[7]:
  - But on any LCD panel, we can take advantage of the known-ordering of the individual R-G-B subpixels that exist within each whole LCD display pixel. By drawing the diagonal edge with some non-white pixels we can produce a much smoother edge to dramatically reduce the jaggies.



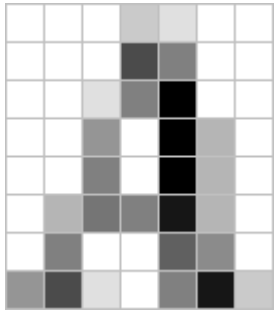
Font Rasterization

Subpixel font rendering

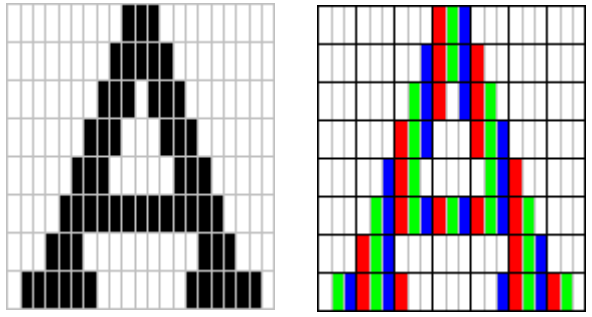
- Here is an another example with font[7]:
  - The task of rendering small typefaces (font) on digital displays is very challenging.
  - For example, 8 point regular capital ‘A’ rendered in Microsoft’s standard Times new Roman font is vastly less clear and sharp than the same character printed on paper.
  - When we zoom-in to see that that capital ‘A’ looks like at the pixel level, we can see the blocky and barely recognizable result.
  - Anti-aliasing technique may be adopted to reduce blocky effect. The hope is that our eyes will tend to average two adjacent gray pixels to see one in the middle.
    - ✓ However, for small type sizes anti-aliasing merely blurs the image and reduces readability.
  - With three times more pixels available horizontally, we can increase the readability of the character.



8 point capital ‘A’



anti-aliasing is adopted



subpixel rendering

Font Rasterization

Subpixel font rendering

- From left to right: no anti-aliasing, subpixel rendering, anti-aliasing, subpixel rendering with anti-aliasing.

Reference

[1] https://en.wikipedia.org/wiki/Typeface
[2] https://en.wikipedia.org/wiki/Baseline_(typography)
[3] https://material.io/design/typography/understanding-typography.html#type-properties
[4] https://en.wikipedia.org/wiki/Body_height_(typography)
[5] https://en.wikipedia.org/wiki/Cap_height
[6] https://en.wikipedia.org/wiki/Font_hinting
[7] https://www.grc.com/ctwhat.htm