This tutorial is “Extension 4” from *Processing: A Programming Handbook for Visual Designers and Artists, Second Edition*, published by MIT Press. © 2014 MIT Press. If you see any errors or have comments, please let us know.

Print

Casey Reas

Digital technologies have spawned many changes to printing within the arts. The introduction of laser printers and personal computers into design offices in the mid-1980s was a catalyst for years of experimentation and innovation in typeface design, layout, and printing. Artists have produced prints from software since the 1960s, but these techniques have surged since 1990. Innovations have given digitally made prints a longer estimated life than color photographs printed from film. The recent deluge of digital cameras provided another change. Amateurs and professionals are skipping the lab and printing their images in the studio and at home.

This short text provides a brief history of the digital printing technologies that have led to these new techniques. It presents examples of software written to produce print output, and discusses a few common contemporary print technologies. The industry surrounding digital printing is full of trademarked names and buzzwords, so this text aspires to demystify some of the terminology and provide pointers to additional information. The content that follows is tailored for printing at home or working with a vendor to produce small editions.

Print and computers

When they originated in the 1960s, computer graphics were more often seen printed on paper than on screens. Computers of this time were enormous, expensive machines that were accessible only at research centers and universities, but prints found their way into galleries, journals, and newspapers. In 1963, the *Computers and Automation* journal announced the first competition for computer graphics to be judged using aesthetic criteria.[1] The U.S. Army Ballistic Missile Research Laboratories won the first two competitions, but A. Michael Noll and Frieder Nake won the following two. In 1965 in Stuttgart, Georg Nees and Frieder Nake were the first individuals to exhibit their computer-generated images in a gallery. The same year, the work of A. Michael Noll and Bela Julesz was exhibited at the Howard Wise Gallery in New York.[2] These shows presented drawings defined by code and output using a plotter. A plotter is a machine that controls the position of a physical pen on a drawing surface. Nake described his plotter in an essay for the Cybernetic Serendipity exhibition catalog: “I used the Graphomat Zuse Z 64 drawing machine controlled by punch tape. The machine has a drawing head guiding four pens, fed by Indian ink of different colours with nibs of varying thicknesses.”[3] Because of mechanical and software limitations, the drawings exhibited in these shows were sparse, mostly geometric, black-and-white images. The plotter remained one of the most common output devices into the 1980s and is still in use today. Over the years, artists have explored many drawing surfaces and have attached brushes, pencils, and other marking instruments to the plotter’s head.

Another area of printed computer graphics produced during the 1960s was more similar to photography than to drawings. At Bell Laboratories, the engineers Kenneth Knowlton and Leon Harmon explored what they called “picture processing.” To create their 1966 Studies in Perception I, a 5 × 12 foot print of a reclining nude, they scanned a photograph with a device similar to a television camera to convert it to a series of numbers stored on a magnetic tape.[4] The picture's range of gray values was reduced to eight levels, and when it was printed using a microfilm plotter, each gray level was replaced with a corresponding icon with a similar density that, when viewed at a distance, simulated the gray value. The icons used for the print included mathematical symbols (multiplication and division signs) and electronics symbols for diodes and transistors. The final enlargement was made from the microfilm using a photographic process. The techniques used to create this work envisioned the now familiar technologies of scanning and image filtering.

During the 1980s, the cost of computers and printing technology fell to levels within reach of individual artists and designers. Mark Wilson started to work with personal computers and plotters in 1980. He utilized the resolution and precision of the plotter to produce dense, geometric textures. He has continued to explore new printing techniques and has produced work using the plotter to deposit acrylic onto linen and to draw with ink on mylar. In his 1985 book *Drawing with Computers*, Wilson explained how to use a personal computer and the BASIC programming language to control a plotter. The following program from the book draws a line from coordinate (100, 100) to (200, 200). The text following each apostrophe is a comment explaining the purpose of each line:

```
100 OPEN "COM1,1200,0,7,1" AS #1              'Serial communications opened
110 PRINT #1,"!AE"                            'Initialize plotter
120 STARTX-100                                'Create and assign STARTX variable
130 STARTY-100                                'Create and assign STARTY variable
140 ENDX-200                                  'Create and assign ENDX variable
150 ENDY-200                                  'Create and assign ENDY variable
160 PRINT #1,"!AX"+STR$(STARTX)+STR$(STARTY);  'Move pen head to coordinate (100,100)
170 PRINT #1,"!AY"+STR$(ENDX)+STR$(ENDY);      'Draw line to coordinate (200,200)
```

Each plotter manufacturer (e.g., Hewlett-Packard, Tektronix, IBM) had its own commands, but they all provided the ability to move the pen up and down and from one location to another. The above example was written for a Tektronix plotter.

The LaserWriter printer, introduced for Apple's Macintosh computer in 1985, was an important innovation in printing technology. Combined with page-layout software and the Mac's GUI interface, this early laser printer was the catalyst for the desktop publishing revolution. The LaserWriter printed at 300 dots per inch (dpi), while the more common dot matrix printers at that time printed at 72 dpi. The PostScript programming language was the essential software component of the LaserWriter. Each printer had a processor that ran the PostScript interpreter to rasterize the data for printing. Forms in a PostScript file are defined by coordinates and shape commands. This makes it possible to transform elements within a composition without losing resolution. The PostScript equivalent of the BASIC program presented above is:

```
/startX 100 def            % Create and assign startX variable
/startY 100 def            % Create and assign startY variable
/endX 200 def              % Create and assign endX variable
/endY 200 def              % Create and assign endY variable
startX startY moveto       % Move to coordinate (100,100)
endX endY lineto stroke    % Draw line to coordinate (200,200)
```

Over time, the PostScript language became the de facto standard for printed output, and it served as a basis for ambitious visual experimentation. Rather than programming PostScript files directly, most people used software like Aldus PageMaker to design pages. Graphic designers started to use this technology to assert more personal control over typographic layout and to explore new visual possibilities.

This expanded freedom was manifested in *Emigre* magazine, started in 1984 with Rudy Vanderlans as editor/designer and his partner Zuzana Licko supplying new typefaces. Emigre is a digital type foundry, and the magazine simultaneously promoted the fonts and served as a protagonist in the wider debate on digital aesthetics within the design community. The pages of *Emigre* were filled with portfolios and interviews with the most interesting designers of that time, including P. Scott Makela, Rick Valicenti, and The Designers Republic. In the time before the Web and blogs, *Emigre* magazine was a place to be informed of new ideas and to discuss possible futures. When the last issue was published in 2005, Rick Poynor, founder of *Eye* magazine, wrote: "*Emigre* emerged at a time when technology was changing design forever, and the magazine sizzled with this energy and excitement." With regard to Vanderlans, Poynor stated, "His page designs were exemplary demonstrations of the new digital design aesthetic."[5]

In *The End of Print* (1995), the design writer Lewis Blackwell wrote, "The designer of today works with resources that did not exist just a decade (or less) ago. The digital age has transformed the tools available and the processes by which ideas are realized. The functionality of earlier production methods has been emulated and superseded by this new technology."[6] The typographic flexibility encouraged by desktop publishing software was pushed to its limit in the pages of *Beach Culture* and *Ray Gun*. David Carson's designs for these magazines about surfing and music were exciting and controversial. The visual style from page to page was wildly idiosyncratic, and the multiple layers of stressed typography and photography often bordered on illegibility, but the design was in the spirit of the content and the time. Carson explored the extremes of letter spacing, typographic texture, and mixing typefaces, but always with a sensitive eye for composition.

Printed work in the 1980s and 1990s was not all created using commercial desktop publishing software. Many artists and designers wrote custom software to realize their vision. The Beowolf typeface designed by LettError utilized code to produce a font that randomizes the design of every letter as it is printed. John Maeda's early printed works from the 1990s fostered a surge of interest in programming images for print. These commercial posters for Japanese printers and type foundries utilize algorithms to generate images of astonishing complexity and resolution. Each of the ten posters for Morisawa use only the company's logo to create a diverse range of dense, delicate typographic compositions. The work of Kenneth A. Huff is more representational than the graphic work of Maeda, yet it too is rooted in algorithmic abstraction. Influenced by organic patterns such as those found in lichen and drying mud, Huff develops unique numerical formulas to use as the basis of his compositions. He places an emphasis on accurate rendering of textures and lighting to evoke tactility, but his work diverges from the constraints of physical materials.

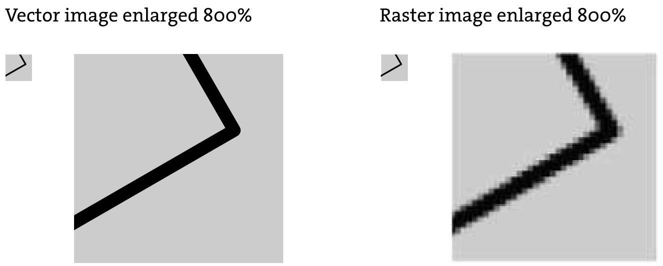
Rapid advancements in inks and paper technology have made it possible for digital prints to have the longevity necessary for the art market. Artists who previously worked with traditional materials and techniques have started to use digital printing technologies in place of etching and lithography, and photographers have replaced darkrooms with computers. In the early 1990s, Iris prints became the first digital prints to be heavily used by established printmakers and photographers. At this time, Robert Rauschenberg, wellknown for his lithographs, began making Iris prints with vegetable dyes and transferring the images to another piece of paper to make collages. This technique was similar to his transferred lithograph images and silkscreen painting from the 1960s, but gave him more control. Other well-known early adopters of digital printing include Chuck Close, Jim Dine, and William Wegman.

Artists have also started to combine new digital techniques with traditional art materials. For example, recent prints from Jean-Pierre Hébert use a computer-controlled machine to etch a copper plate, and the physical printing process is executed traditionally. Manfred Mohr prints onto canvas and then stretches the prints over frames. Subsequent technologies have again changed the landscape of digital printing, and more artists continue to use the technology.

High-resolution file export

Images saved directly from screen are created at the screen's resolution, typically around 100 pixels per inch (ppi). This low resolution is clearly visible when the images are printed with a high-resolution printer. In contrast to screen resolution, printers are capable of 9600 dpi. Two primary techniques are used to create high-resolution files with software. The first technique saves a vector file, and the second saves a raster (bitmap) file. The vector technique creates files that store shape information as coordinate points. Common vector formats are PDF, AI, EPS, and SVG. The raster technique stores shape information as pixels. Common raster formats are TIFF, JPEG, TARGA, and PNG. A vector file can be output at any size without loss of resolution, but raster files do not scale gracefully.

The difference is illustrated with this diagram:



Vector image enlarged 800%

Raster image enlarged 800%

Use the vector technique to export line art, type, or shapes that can be printed professionally, published, or printed at very large sizes. It's also helpful to create a file that can be edited further with a program like Inkscape or Adobe Illustrator. Raster images are useful when exporting an image from a program that does not refresh its background each frame. If the image accumulates by adding each new frame to the display window, it may not be possible for vector data to achieve the same effect, or it may be too much geometry to store in a single file. A raster file does not represent each visual element separately (it saves it as a series of pixels), so it is editable only by programs like GIMP and Photoshop. A raster file can be printed with as much resolution as a vector file if it is output with a large enough width and height setting to give the file a high resolution when scaled for print. For example, to print a four-inch image at 600 dpi would require size(2400,2400) inside setup(). Vector files are eventually rasterized during the printing process, so it's simply a matter of when the rasterizing takes place—whether directly from the program, or inside a professional raster image processor (RIP). After a file is generated through software, it is often modified before it is printed. Common changes include tweaking color or changing the weight of the lines after print tests. To make changes, load raster files into a program such as GIMP or Photoshop. Load vector files into a program such as Inkscape, Illustrator, or CorelDRAW. The following examples clarify the strengths and weaknesses of each technique.

Example 1: Render to PDF

When PDF is used as the third parameter to the `size()` function, the program renders to a PDF file instead of drawing to the display window. The file name is set by a fourth parameter to `size()` and the file is saved to the sketch's folder. Most sketches can be rendered as PDF by simply adding the two parameters to the `size()` command and selecting Sketch -> Import Library -> PDF. Once you do this, you'll no longer see the image on-screen as it is running, but it becomes possible to create PDF files at sizes much larger than the screen.

```
/*
 * Example 1: Render to PDF
 */

import processing.pdf.*;          // Import PDF code

size(600, 600, PDF, "line.pdf");  // Set PDF as the renderer
background(255);
stroke(0);
line(200, 0, width/2, height);    // Draw line to PDF
exit();                           // Stop the program
```

Example 2: Render to screen, export to PDF

This example saves a PDF file while simultaneously drawing to the screen. The `beginRecord()` function opens a new file, and all subsequent drawing functions are echoed to this file as well as to the display window. The `endRecord()` function stops the recording process and closes the file. The `beginRecord()` function requires two parameters; the first is the renderer to use (in this example, PDF), and the second is the file name.

```
/*
 * Example 2:  Render to screen, export to PDF
 */
```

```
import processing.pdf.*;          // Import PDF code

size(600, 600);
beginRecord(PDF, "line.pdf");     // Start writing to PDF
background(255);
stroke(0, 20);
strokeWeight(20);
line(200, 0, 400, height);        // Draw line to screen and to PDF
endRecord();                      // Stop writing to PDF
```

Example 3: Save one frame from a continuous program

This example saves a PDF file each time the mouse is pressed. The boolean variable saveOneFrame is set to true when a mouse button is pressed, causing beginRecord() to run the next time through draw(). At the end of draw(), endRecord() is run and the variable is set to false so another file won't be saved while drawing the next frame. Each PDF file is numbered with the current frame (the number of elapsed frames since the program started).

```
/*
 * Example 3:  Save one frame from a continuous program
 */

import processing.pdf.*;  // Import PDF code

boolean saveOneFrame = false;

void setup() {
  size(600, 600);
}

void draw() {
  if (saveOneFrame == true) {              // When the saveOneFrame boolean is true,
    beginRecord(PDF, "line-####.pdf");    // start recording to the PDF
  }
  background(255);
  stroke(0, 20);
  strokeWeight(20);
  line(mouseX, 0, width-mouseY, height);
  if (saveOneFrame == true) {              // If the PDF has been recording,</em>
    endRecord();                           // stop recording,
    saveOneFrame = false;                  // and set the boolean value to false
  }
}

void mousePressed() {                      // When a mouse button is pressed,
  saveOneFrame = true;                     // trigger PDF recording within the draw()
}
```

Example 4: Accumulate many frames into one PDF

This example saves multiple frames drawn to the screen within a single PDF file. The file opens when the *B* key is pressed, and everything drawn in subsequent frames is saved into it, until the *E* key is pressed. The background function is run after beginRecord() to clear the background within the PDF document as well as in the display window. This example draws only one new line to the PDF file each frame so the file remains small, but it's possible to write thousands of lines each frame. However, when vector files get very large, computers can have difficulty opening and printing them.

```
/*
 * Example 4:  Accumulate many frames into one PDF
 */

import processing.pdf.*;  // Import PDF code

void setup() {
  size(600, 600);
  background(255);
}

void draw() {
  stroke(0, 20);
  strokeWeight(20);
  line(mouseX, 0, width-mouseY, height);
}

void keyPressed() {
  if (key == 'B' || key == 'b') {         // When 'B' or 'b' is pressed,
    beginRecord(PDF, "lines.pdf");        // start recording to the PDF
    background(255);                      // Set a white background
  } else if (key == 'E' || key == 'e') {  // When 'E' or 'e' is pressed,
    endRecord();                          // stop recording the PDF and
    exit();                               // quit the program
  }
}
```

Example 5: High-resolution off-screen buffer

This example creates a TIFF file larger than the screen and draws into it directly, rather than drawing to the screen. The `createGraphics()` function creates an object from the `PGraphics` class (`PGraphics` is the main graphics and rendering context for Processing). The `beginDraw()` method is necessary to prepare for drawing, then each subsequent drawing function is written into the large raster object. The `endDraw()` and `save()` methods are necessary to complete the file and then save it to the machine so that it can later be viewed in a different program such as GIMP or Photoshop.

```
/*
 * Example 5: High-resolution off-screen buffer
 */

PGraphics big;  // Declare a PGraphics variable

void setup() {
  big = createGraphics(3000, 3000);  // Create a new PGraphics object
  big.beginDraw();                   // Start drawing to the PGraphics object
  big.background(128);               // Set the background
  big.line(20, 1800, 1800, 900);     // Draw a line
  big.endDraw();                     // Stop drawing to the PGraphics object
  big.save("big.tif");
}
```

Example 6: Scale and segment an image

This example saves a series of image files at screen resolution from a single image enlarged to any dimension. These files can be tiled within an image editor such as Photoshop to create a single, high-resolution file. Another program can also be written to tile the images together automatically. A `scaleValue` of 2 tiles the image to 4 files, a `scaleValue` of 3 tiles the image to 9 files, etc. This example code works only with 2D images.

```
/*
 * Example 6: Scale and segment an image
 *
 * Draws an image larger than the screen by tiling it into small sections.
 * The scaleValue variable sets amount of scaling: 1 is 100%, 2 is 200%, etc.
 */

int scaleValue = 3;  // Multiplication factor
int xoffset = 0;     // x-axis offset
int yoffset = 0;     // y-axis offset

void setup() {
  size(600, 600);
  stroke(0, 100);
}

void draw() {
  scale(scaleValue);
  translate(xoffset * (-width/scaleValue), yoffset * (-height/scaleValue));
  line(10, 150, 500, 50);
  line(0, 600, 600, 0);
  setOffset();
}

void setOffset() {
  save("lines-" + xoffset + "-" + yoffset + ".jpg");
  xoffset++;
  if (xoffset == scaleValue) {
    xoffset = 0;
    yoffset++;
    if (yoffset == scaleValue) {
      exit();
    }
  }
  background(204);
}
```

Printing technologies

Like traditional printing technologies, creating a high-quality digital print is a craft that requires knowledge and experience in addition to excellent tools and machines. The quality of a print is affected by the quality of the printer, ink, and paper, the preparation of the digital file, and the printer settings. Each of these components is introduced below. Many different printing technologies are currently in use by artists and designers, and each has unique attributes. This list presents some of the most popular ones. Because printing technology changes rapidly, specific printer models are not discussed.

Laser

Laser printers are exceptional because of their high resolution, speed, and low cost per page. For these reasons, they are ubiquitous in office environments. Laser printers use a technology similar to that of photocopiers. When a print is made, a cylindrical drum inside the printer is electrically charged. A high-precision laser is reflected to strike the drum, and it reverses the electrical charge where it hits. When the drum comes in contact with charged toner (small particles of carbon blended with a polymer), the toner is attracted to the drum where the laser hit. A sheet of paper is then passed over the drum, and the toner is transferred and then fused to the paper surface with heat.

Inkjet

Inkjet prints are the most common technology for home printers because of their low cost and high image quality. They have also become a dominant technology for professional photographic printing. In comparison to laser printers, inkjet printers are slow, but they can produce images with a much higher resolution. They achieve this resolution by precisely squirting tiny droplets of ink onto the page. For this reason, the actual resolution of an inkjet can't compare directly to that of a laser printer, which is more precise. A 600 dpi inkjet printer may produce distinguishable shapes down to 150 dpi, while a laser printer at 600 dpi maintains precision almost to the full 600 dpi. Each inkjet printer operates using one of three technologies: thermal, piezoelectric, or continuous. The thermal technique forces a drop of ink onto the paper by using heat to cause a tiny steam explosion within a chamber. The piezoelectric technique bends a piezoelectric crystal to force a droplet out of the ink chamber. The continuous technique sends a constant flow of droplets but charges each with a varying electrostatic field that determines how or whether it will hit the paper. Inkjet printers are sometimes also called bubblejet printers.

Digital chromogenic print (C Print)

A digital C print is similar to a traditional color photographic print, but a digital file is used instead of a negative. A C print is made by exposing photographic paper to light (either an LED or laser) inside the printer. It is then processed using chemicals, just like a photographic print. Chromira and Lightjet are two popular types of printers that use this technique.

Iris

Iris printers were developed to make full-color proofs before a job was printed in large numbers on a commercial press. They started to be used as fine art printers in the early 1990s. The most distinct aspect of this technology is the ability to print on many different flexible substrates including paper, silk, and canvas. To make an Iris print, the substrate is attached to a metal drum and spun at a high speed while tiny drops of ink are applied to the substrate in a high-pressure stream. An Iris printer is a specific type of inkjet printer.

Giclée

Giclée (pronounced zhee-CLAY) is not a specific printing technology; it is a term used to define a high-quality, digitally produced fine art print. The term giclée was selected to distance the technique from connotations to digital technology and computers, as a tactic to gain acceptance within the art community. The first giclée prints were made using the Iris printing technology, but they are now made using other technologies as well.

Other techniques

In addition to the printing technologies mentioned above, commercial printers offer a wide selection of specialized printers for creating large-format prints for buses, billboards, and buildings. These printing techniques produce lower-resolution images, but they look crisp at a distance.

Ink

The various printing technologies use different types of inks. Laser printers use toner, inkjet printers use liquid inks, and C prints don't use inks because they are made with a photographic process. Toner must be fixed with heat, but liquid inks are absorbed into the paper. Inkjet inks can be divided into two broad categories: dye-based and pigmented. In comparison to pigmented inks, the more common dye-based inks are not water-resistant, are less expensive, are less resistant to fading, and can create more vivid colors.

Ink selection partially determines how long a print will last. Inkjet inks have been notorious for fading color (early prints could fade within six months), but prints from some of the current generation of printers are rated to hold their color for over 100 years when used with special papers. Ultraviolet (UV) light is a huge contributing factor to fading. Use frames with UV-filtering glass or plexiglass to significantly increase the life of the print.

Paper

Paper is defined by its surface, material, and weight. Basic surface options include matte, luster, semigloss, gloss, and supergloss with textures ranging from extremely smooth to rough. Paper is typically made from wood fibers, but fibers from cotton, hemp, linen, and rice are also used. Paper made from wood is naturally acidic, so look for papers that are acid-free (pH neutral) to increase the longevity of prints. Paper made from 100 percent rag is the most stable and will not grow brittle with age. The weight of a paper affects its thickness. Fine art papers are usually measured in units of grams per square meter (gsm or g/m2). Common weights range from 110 (thin) to 350 (thick).

The selection of papers available for digital printing is extremely limited in comparison to those manufactured for traditional fine art and commercial printing, but the selection is still broad. Printer manufacturers such as Hewlett-Packard, Canon, and Epson offer their own papers, but the finest-quality paper can arguably be found at companies that specialize in making paper. Companies like Hahnemühle and Somerset have recently started producing papers specifically for digital printing technologies. Some inkjet printers require paper to be coated with a special layer to stop the ink from bleeding into the paper and dulling the color. Iris printers offer the widest selection of printable media. An Iris printer can print on any absorbent material and still produce strong color.

Paper for printers comes in sheets and rolls, but depending on where you live, it will either be sized according to the international ISO 216 standard (A4, A3, etc.) or North American sizes (letter, tabloid, etc.)

File preparation

The format, resolution, and color profile are the most important components of preparing a file. For the best printing results, files should be saved in a format that does not compress the data in a way that loses information. For instance, the JPEG image format compresses a file by removing color data. Each type of printer produces the best results when files are at a specific resolution. Images should typically be saved at 300 dpi or higher. Images

Cover

Download

Exhibition

Reference
Libraries
Tools
Environment

Tutorials
Examples
Books
Handbook

Overview
People

Shop

» Forum
» GitHub
» Issues
» Wiki

» FAQ
» Twitter
» Facebook

produced for Inkjet printers should be prepared at a dpi resolution that is an increment of the maximum printer resolution. For example, an image file prepared for a 2880 dpi printer should be saved at 360 dpi (360 × 8 = 2880). The 360 dpi resolution is suggested for photographic images, but the quality of an image with fine lines can be improved by doubling the resolution to 720 dpi (720 × 4 = 2880). A higher dpi resolution won't help a great deal (with inkjet) and will significantly increase file size. The color profile for a file is essential to match color when working with other people or across different computers. Each file should be tagged with a color profile to specify the color space of the document (e.g., ColorMatch RGB or Adobe RGB 1998). A color profile tells an output device such as a monitor or printer how to interpret the file's numerical color data to display it correctly on that device.

Our experience has shown that complex vector files should be rasterized in a program such as Adobe Photoshop before they are sent to a vendor for printing. Most printers specialize in printing photographs and have more competence in working with image formats like TIFF and PSD. Printers eventually rasterize the image, and it's a good idea to have complete control over this process, unless the print shop specifies otherwise.

Inkjet printer settings

C prints and Iris prints are typically made through a vendor and laser printing is straightforward. Making a high-quality print with an inkjet printer, however, is often done at home but requires following a few important protocols. Because every type of paper behaves differently with each printer, it's necessary to define the paper type within the print dialog box. Most printers offer selections for their proprietary papers. If you are using one of these papers, select it from the list. If you are using a different paper, you may want to install the ICC profile for the paper that matches the printer.[7] The dialog box will also offer an option to set the dpi of the print. Sometimes it's necessary to go into the advanced settings to gain access to this. There are a few things to consider when selecting the dpi. Glossy papers can hold a higher resolution than matte papers, and higher-resolution prints require more ink. Unless your print has extremely precise details and will be viewed close up, 1440 dpi is an adequate resolution.

Making test prints is an essential step toward producing a high-quality print. Precise colors and line weights look very different on-screen than on paper. If working with a vendor, always have a small test print made to check the quality before producing the final print.

Conclusion

Only within the last fifteen years have digital printing technologies begun to rival traditional printing techniques in their resolution and longevity, but there is a vast discrepancy between the physical qualities of a digital print and an original Hokusai (1760–1849) woodblock print or an Albrecht Dürer (1472–1528) etching. Each printing technique has its constraints and advantages. One arguable advantage of digital printing is the absence of a physical representation of the print. A lithograph has a corresponding stone, and offset printing has a metal plate, but the information for a digital print is stored in a computer's memory and is therefore easier to modify. The primary advantage of digital printing is the ease with which a software image can be manifested as a physical image. In contrast to an image on-screen, a print can have a much higher resolution and will exist long after the software becomes incompatible with future computers.

There are also disadvantages to printing software images. The ability to animate the image is lost, and the color palette is reduced. Digital printing technology evolves rapidly, and it's possible that the current research into electronic paper (e-paper, e-ink) and organic light-emitting diodes (OLEDs) will allow the best aspects of printed images to merge with the best elements of digital screens. These or other emerging technologies may make it possible for digital printing to evolve into an original area of image-making rather than simply mimicking traditional printing technologies.

Notes

[1] H. W. Franke, *Computer Graphics, Computer Art* (Phaidon, 1971), p. 60.

[2] Ibid., p. 69.

[3] Frieder Nake, “Notes on the Programming of Computer Graphics,” In *Cybernetic Serendipity*, edited by Jasia Reichardt (Praeger, 1969), p. 77.

[4] This work is also credited as *Mural* in the exhibition catalog *Cybernetic Serendipity*.

[5] Rick Poynor, “*Emigre*: An Ending,” Design Observer, 10 November 2005.
http://www.designobserver.com/archives/007816.html.

[6] Lewis Blackwell, *The End of Print: The Graphic Design of David Carson* (Chronicle Books, 1995), p. 173.

[7] ICC profiles are files that define the mappings between a print’s data and the specific paper and printer specifications. Well-designed profiles can increase the quality of a print. Check the website of the paper manufacturer to see if it has created one for your printer.