Module IV: Art and Images

Computer Art: A Brief History

Possible Questions for Art Critique*

- **description**: What do I see?
- **medium**: What tools, materials, or processes did the artist use?
- **form**: What elements did the artist choose and how did the maker organize the elements?
- **interpretation**: What is the artwork about?
- **criteria**: What criteria do I think are most appropriate for judging the artwork?
- **evidence**: What evidence inside or outside the artwork relates to each criterion?
- **judgment**: Based on the criteria and evidence, what is my judgment about the quality of the artwork?

* based roughly on Terry Barrett's *Criticizing Art: Understanding the Contemporary* (1994)
(Pre-)Computer Art 1940s and before

- Jacquard loom wove patterns according to instructions encoded in punched cards (early 1800s)
- General-purpose computing devices
  - Analytical Engines—envisioned by Charles Babbage, Ada Lovelace (mid 1800s)
- First electronic computer, the ENIAC, completed in 1946, motivated by goals of mathematical calculations and military research
(Pre-)Computer Art 1940s and before

“The Analytical Engine […] is the introduction into it of the principle which Jacquard devised for regulating, by means of punched cards, the most complicated patterns in the fabrication of brocaded stuffs […] We may say most aptly that the Analytical Engine weaves algebraical patterns just as the Jacquard-loom weaves flowers and leaves.” – Ada Lovelace

(Pre-)Computer Art 1940s and before

“The Analytical Engine might act upon other things besides number, were objects found whose mutual fundamental relations could be expressed by those of the abstract science of operations […] the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent” – Ada Lovelace

Early 1950s

- computers were used primarily for mathematical calculation
- IBM concludes that five computers are sufficient for U.S. market
- little access to computers for artists

Mid 1950s – Early 1970s

- computer-driven pen plotters
  - vector graphics
  - incapable of filling areas
  - “technical”-looking

- images were produced using line-drawing algorithms
  - access, programming expertise needed
  - works could not be photo-realistic
Mid 1950s – Early 1970s

• much interest at Bell Labs, a research lab in New Jersey (USA)

• first computer graphics group at Boeing (why Boeing – an airplane manufacturer?)

• early examples of computer animation

A. Michael Noll’s Experiments

• researcher (engineer, computer scientist, artist) at Bell Labs

• used computers to explore roles of randomness and chance in “laws of aesthetics”

• used random number generators to add variation to drawings

• landmark studies compared peoples’ preferences for computer generated “Mondrian-like” drawings with Mondrian original

A. Michael Noll

Piet Mondrian
Vera Molnar’s Early Work

- trained as an artist in Budapest, worked in Paris
- interested in exploring, in a systematic way, how distortions to basic geometric elements could lead to aesthetically appealing work

“Without the aid of a computer, it would not possible to materialize quite so faithfully an image that previously existed only in the artist's mind. This may sound paradoxical, but the machine, which is thought to be cold and inhuman, can help to realize what is most subjective, unattainable, and profound in a human being.” - Vera Molnar

Vera Molnar’s Early Work

“Vera Molnar holds that the computer can serve four purposes. The first concerns its technical promise - it widens the area of the possible with its infinite array of forms and colors, and particularly with the development of virtual space. Secondly, the computer can satisfy the desire for artistic innovations …. Thirdly, the computer can encourage the mind to work in new ways. Molnar believes that artists often pass far too quickly from the idea to the realization of the work. The computer can create images that can be stored for longer... Finally, Molnar thinks that the computer can help the artist by measuring the physiological reactions of the audience, their eye movements for example, thus bringing the creative process into closer accordance with its products and their effects.” – Frank Popper
Raster Graphics (1970s)

- technology:
  - screen is a 2D array (grid) of phosphor dots, turned on/off by a beam that sweeps the screen
  - beam sweeps row by row (raster by raster)
  - areas can be filled in, unlike plotter-like systems

- impact:
  - image representation changed (vector to raster)
  - enabled visually realistic interactive graphics
  - technology accessible beyond programmers
  - capabilities to flip, scale, juxtapose pictures
  - new ways to experiment with colour

Advent of GUIs (and new input devices)

- Sketchpad (Sutherland, 1960s) allowed images to be drawn interactively with a light pen
- innovations at Xerox PARC in early 1970s led to GUIs

1970s – 1980s

- use of computer graphics in fields of design, architecture
- experiments with photographic images
- use of computers in art analysis
- still expensive, also some stigma
Joan Truckenbrod

- trained as an artist in the U.S.
- line-drawing art in 1970s
- used early Apple computer to generate algorithmic images that were transferred to textiles
- was a pioneer in the manipulation of photographic images to "combine realities"

“...Coding was a vehicle for giving form to algebraic formulas that scientifically describes invisible forces such as how light waves reflect off of irregular surfaces. These symbols reappear in my later images as there is a mapping of mathematical symbols onto a natural ecology, fusing with layers of leaves and flower petals. Unfolding in the images is an architecture of social constructs."

– Joan Truckenbrod

Harold Cohen’s AARON

- Harold Cohen, a successful English painter, started to experiment with computers in 1968
- interested in understanding “cognitive rules” of image-making and painting; what is creativity
- led to development of AARON, an artificial intelligence (AI) system for painting
Early drawing usually proceeds out of the purely motor activity we call scribbling. At some point a round-and-round scribble will migrate outwards and become an enclosing form, and at some time rather later, the scribble will be omitted, leaving closed forms that are then available for the building of a variety of representational elements…"

– Harold Cohen

http://crca.ucsd.edu/~hcohen/cohenpdf/colouringwithoutseeing.pdf
Harold Cohen’s AARON

“I came to computing after a reasonably successful twenty-year career as a painter, at a stage in which I was feeling increasingly frustrated by my own lack of understanding of how painting “works” [...] My need wasn’t to make art — I knew very well how to do that — but to understand what art is [...] It is nonsense (left-brain, right-brain bla-bla) to say that programming is too difficult for artists to learn. Anyone can learn anything if they really need to be able to do it. And if the need is powerful enough it really doesn’t matter how long it takes. It’s a sight harder to learn programming when you’re forty, as I was, than when you’re twenty — and it was a sight harder thirty-five years ago than it is today. – Harold Cohen

1990s - Present

- computers increasingly affordable and powerful
- many new technologies
  - 3D graphics and animation tools (spurred by movie industry)
  - world wide web
  - immersive (CAVE) environments
- many new art forms
  - artificial life, artificial worlds
  - internet art: building on internet medium
  - interactive art
- steep learning curve for many new technologies

Projects

- Holger’s Art: www.cs.ubc.ca/~hoos/art.html
- Sid Fels, ECE, UBC
  - several projects that explore human-computer interaction through art, see www.ece.ubc.ca/~ssfels/introduction.html
- graphics projects at UBC/CS: www.cs.ubc.ca/labs/imager/imager-web/Research/graphics.html
- graphics projects at U. Washington: grail.cs.washington.edu/projects
Connecting with Computer Science

Resources

• “The Computer in the Visual Arts” by Anne Morgan Spalter
• Leonardo: Journal of the International Society for the Arts, Sciences and Technology
• Internet Art Database: dart.fine-art.com
• Digital Art Museum: www.dam.org (source for most images presented in these slides)
• SIGGRAPH art gallery: www.siggraph.org/artdesign

Connecting with Computer Science

Module IV:
Art and Images

Image Representation

Data Representation

Exercise: Pick two examples of data in everyday life; one computer-related one not.

• In what media are the data represented?
• Do you convert the data from one representation to another, as you use the data? How?
• How does a particular representation of the data influence its utility (e.g., what you can do with it)?

Data Representation

• Representing data in different ways, using different media, is a fundamental human activity.
• Computers can help!
• Let’s look at data representation in more detail, focusing for now on representation of images on a computer
• Computers use digital representations, using the bits 0 and 1
Learning Goals [for 2+ classes]

you should be able to

- define the RGB colour specification, explain its basis, and convert from hexadecimal to decimal
- define “bitmap image” and “pixel” and explain how to construct a bitmap image representation
- define “vector image” and explain how to construct a vector image representation
- compare and contrast the suitability of bitmap and vector representations for different uses of images

Clicker Exercise

Suppose red’s intensity is 255 (full intensity). What happens if both the blue and green intensities increase at the same rate, starting from 0?

A. The colour remains red but gets lighter
B. The colour remains red but gets darker
C. The colour changes from red to an aqua shade
D. None of the above

RGB Colour Representation

- Cones in the human visual system are of three types, which perceive red, blue, and green light.
- We perceive other colours as mixtures of these three.
- No more than 200 intensities of a given colour are distinguishable by the human eye.

- Colours can be specified as $RGB$ (red-blue-green) intensities.
- Computer applications use 256 intensities
- 256 possibilities can conveniently be stored in 8 bits, i.e., one byte, of computer storage, using binary notation.
Specifying Colour Intensities

- decimal: three numbers from 0 … 255
  - e.g. 0, 5, 255
  - human friendly
- binary: three 8-bit numbers
  - e.g. 00000000, 00000101, 11111111
  - computer friendly
- hexadecimal: three 2-digit numbers, where digits are 0,1,…9, A, … F
  - a human-computer compromise
  - used, e.g., in HTML specifications

**goal: be able to convert from hexadecimal to decimal**

### Conversion Table: Small Numbers

**Exercise: fill in the gaps**

<table>
<thead>
<tr>
<th>dec</th>
<th>bin</th>
<th>hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>0000</td>
<td>00</td>
</tr>
<tr>
<td>01</td>
<td>0001</td>
<td>01</td>
</tr>
<tr>
<td>02</td>
<td>0010</td>
<td>02</td>
</tr>
<tr>
<td>03</td>
<td>0011</td>
<td>03</td>
</tr>
<tr>
<td>04</td>
<td>0100</td>
<td>04</td>
</tr>
<tr>
<td>05</td>
<td>0101</td>
<td>05</td>
</tr>
<tr>
<td>06</td>
<td>0110</td>
<td>06</td>
</tr>
<tr>
<td>07</td>
<td>0111</td>
<td>07</td>
</tr>
<tr>
<td>08</td>
<td></td>
<td>08</td>
</tr>
<tr>
<td>09</td>
<td>1001</td>
<td>09</td>
</tr>
<tr>
<td>10</td>
<td>1010</td>
<td>0A</td>
</tr>
<tr>
<td>11</td>
<td>1011</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1100</td>
<td>0C</td>
</tr>
<tr>
<td>13</td>
<td>1101</td>
<td>0D</td>
</tr>
<tr>
<td>14</td>
<td>1110</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>1111</td>
<td>0F</td>
</tr>
</tbody>
</table>

*leading 0’s can be omitted*

### From Hex to Binary and Back

- easy: replace each hexadecimal "digit" with the corresponding four binary digits using the conversion table
- examples:

<table>
<thead>
<tr>
<th>hex</th>
<th>binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC</td>
<td>10101100</td>
</tr>
<tr>
<td>11</td>
<td>01101001</td>
</tr>
<tr>
<td>FF</td>
<td>11111111</td>
</tr>
</tbody>
</table>

### Colour and Number Representation

**summary: colour representation**

- colours can be represented as RGB intensities—three numbers
- HTML specifications express the RGB numbers in hexadecimal notation
  - example: `<body bgcolor="#3A00FF">`
Colour and Number Representation

**summary: number system calculations**

- **hex → binary**: replace each hexadecimal digit by the four corresponding binary digits in the conversion table
- **binary → hex**: put extra 0’s at the left of the binary number as necessary so that the total number of digits is a multiple of 4, then reverse the hex → binary conversion process

**Colour and Number Representation**

- **adding 1 to a binary number**: replace the rightmost 0 by a 1 and any 1’s to the right of that by a 0
  - example: $01011011 + 1 = 01011100$
- **binary → decimal**: for each place $i$ in the binary number, multiply the digit (0 or 1) in that place by the place value $2^i$ (see next slides)

**From Binary to Decimal**

**first recall meaning of decimal notation**

<table>
<thead>
<tr>
<th>2 4 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1s place multiply by 1 $(10^0)$: 8</td>
</tr>
<tr>
<td>10s place multiply by 10 $(10^1)$: 40</td>
</tr>
<tr>
<td>100s place multiply by 100 $(10^2)$: 200</td>
</tr>
</tbody>
</table>

Total in decimal (add them up): 248

**From Binary to Decimal**

**adapt this meaning to binary notation**

<table>
<thead>
<tr>
<th>1 1 0 0 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1s place multiply by 1 $(2^0)$: 1</td>
</tr>
<tr>
<td>2s place multiply by 2 $(2^1)$: 0</td>
</tr>
<tr>
<td>4s place multiply by 4 $(2^2)$: 0</td>
</tr>
<tr>
<td>8s place multiply by 8 $(2^3)$: 8</td>
</tr>
<tr>
<td>16s place multiply by 16 $(2^4)$: 16</td>
</tr>
</tbody>
</table>

Total in decimal (add them up): 25
Clicker Exercise

The 8-bit binary representation of 57 is 00111001. What is the 8-bit binary representation of 58?

A. 01011110
B. 00111111
C. 00111010
D. 00011101

Clicker Exercise

The largest 2-digit hexadecimal number is

A. 1F
B. FF
C. F1
D. 11

Learning Goals [revisited]

you should be able to

• define the RGB colour specification, explain its basis, and convert from hexadecimal to decimal
• define “bitmap image” and “pixel” and explain how to construct a bitmap image representation
• define “vector image” and explain how to construct a vector image representation
• compare and contrast the suitability of bitmap and vector representations for different uses of images

Representing Images

Exercise:

• How might you verbally describe the following images?
• Could someone accurately reproduce the images based on your description?
Bitmap Image Representation

• to fully specify an image using bitmap representation you need:
  – width and height of the image
  – number of rows and columns of the grid
  – the pixels, i.e. colour intensity values of each grid cell

Compressing Bitmap Images

two techniques used by .jpg files

• areas with similar colour can be modified to have the same colour
• "runs" of identical intensities can be collapsed
Compressing Bitmap Images

**collapsing runs: example**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>G</td>
<td>G</td>
</tr>
<tr>
<td>G</td>
<td>G</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

- Cells are ordered from top left to bottom right: [1,1],[1,2],...,[1,6],[2,1],...,[2,6]
- Ordered list of four “maximum-length” runs:
  1. \((9,33,25)\) \(1\) \((221,192,127)\) \(3\)
  2. \((9,33,25)\) \(4\) \((221,192,127)\) \(4\)
- Can the matrix be reconstructed from the list of runs alone?

Vector Image Representation

**basic approach**

- describe each object of the image either as
  - a sequence of dots (connected by lines), or
  - a simple shape (e.g., rectangle, circle)
- describe the colour (fill) of each closed object

**example**

- the representation is a sequence of numbers
- it should be possible to unambiguously reconstruct the image from the representation
Clicker Exercise

• which image is represented by the following sequence?

   1, 2, 1, 2, 3, 3, 2, 1, 2, 2, 1

A. ![Image A]
B. ![Image B]
C. ![Image C]

---

Clicker Exercise

• which are vector representations?

A. tin toy & pig  C. tin toy & Seurat
B. pig & Seurat  D. just tin toy

---

Clicker Exercise

• which are bitmap representations?

A. rabbit & needlepoint  C. needlepoint & fonts
B. just needlepoint  D. just fonts

---

Vector Image Representation

extensions of basic approach

• add width and colour of lines
• use curves instead of lines
• add shading to objects
• specify which objects overlay others
• extend to 3D
• …

---

Connecting with Computer Science | cs.ubc.ca/~hoos/cpsc101
Exercise: Write a Bitmap Generator!

- your algorithm should describe how a flatbed scanner generates a bitmap representation of an image placed on its glass pane
- see handout

Bitmap or Vector Representation?

*do these devices produce vector or bitmap representations?*

- cutting table
- photocopier
- pen plotter

Bitmap vs Vector Representation

*what properties distinguish these data types?*

- **size**: bitmap representations tend to be bigger than vector representations (even when compressed)
- **editing options**: (see notes)
- **generality**: bitmap representation can be applied to *any* type of image; this is not really true of vector representation
- **scalability**
Exercise

choose dots to represent the following images:

• How to choose the number of dots to use?
  Where to position the dots?
• What principles would you suggest in general, for selecting the dots to represent a picture?