Ready ? Fix : Battery
American Memory in 1/30 Second

Chris Allert

July 17, 2004

Television and film simulate continuous motion by showing images in rapid succession, one still image at a time. On an American TV set, the image usually changes 29.97 times per second. For some time I have wondered what, other than simulating motion, can be done in about 1/30 of a second. This project uses this brief unit of time, the single frame of NTSC video, to explore what we as a people remember.

1 The Setting

The setting is a dark room. The only source of light is five 13 inch TV sets about two feet off the floor, arranged in an arc, with each TV set about two feet from its neighbors to the left and right. On each TV set a different program is running. On the floor about a dozen small mats are arranged indicating good observation points.

2 The TV Programs

The TV programs all have the same structure. For each frame of the program, there is either an image and a short sound or blackness and silence. If there is an image, it is always different from the last image. For each program, there is a predefined period of blackness and a predefined period of images. So each program will flicker between something and nothing at a fixed frequency. The duration of blackness (the "nothing" part of the program's cycle) for a program is between zero and 28 frames (almost one second). The duration of images (the "something" part of the cycle) is between one and seven frames. On average, there will be almost 4 times as much blackness as images. If a program has the maximum duration of blackness (28 frames), the images part of the cycle will be only one frame. Conversely, if a program has the maximum duration of images (seven frames), the duration of blackness will be zero. This pseudocode will clarify how the durations are chosen for a program (variable names in italics):

1. Set *images* to random real number between 0 and 7
2. Set *blackness* to random real number between 0 and *images*
3. Subtract *blackness* from *images*
4. Multiply *blackness* by 4
5. Convert *blackness* and *images* to integers rounding downward
6. Set *images* to 1 if it is 0

Although this algorithm looks somewhat arbitrary, it is the result of a long period of trial and error.

In addition to each TV set showing a different program, the program on each TV set changes every few seconds. At any time, there are between one and nine programs that a TV set is changing between. From this point forward I refer to this set of programs as the *available programs* for a TV set. Every time the program changes, some programs in the set of available programs may be discarded, and some new programs may be added to the set of those to choose from. Once a program is discarded, it is never shown again. Each TV set maintains its own set of available programs and each program is never shown on more than one TV set.

In addition to a duration for images and a duration for blackness, a program also has a collection of between one and thirteen photos from which it chooses one to display at a time. There are 29,957 *source photos* in this project. So when a new program is created, it chooses between one and 13 photos from this collection, removing them from the list of photos available to future programs. When a program displays an image in a frame, it chooses one of its photos at random, crops a random region from that photo, and displays it at a random location on the TV screen in a random color. There are some constraints on the size of the cropped region and the color.

Here is how the cropped region of the photo is determined:

1. All of the source photos are either 600 pixels wide or 600 pixels high. The other dimension is less than 600 pixels.
2. The TV screen is 640 pixels wide and 480 pixels high.
3. Half the time the program will scale the photo to twice as wide and twice as high before cropping it.
4. The program will choose a cropping rectangle that is between half and full screen width wide and half and full screen width high.
5. If the photo is shorter or narrower than the screen it will choose between half and full of the image's dimension instead of the screen's dimension.
6. After the size of the cropping rectange has been set, it will be placed at a random location within the source photo and a random location within the TV screen will be chosen for it to appear.

And here is how the image is colored when it is displayed:

1. Every source photo has three *color channels*; a *red*, a *green*, and a *blue* one.

2. The program will choose between one and all of these to display.

3. For each channel to be displayed it will be displayed in a random *rainbow color* between half and full brightness.

4. A rainbow color is two colors added together. One is red, green, or blue at full intensity, the other is red, green, or blue (but not the same color as the full intensity color) at any intensity.

The above process is analogous to taking a black and white photo of a color photo through a red, green, or blue filter, and then printing that photo on bright colored paper, then dimming the lights as you look at the result.

In addition to displaying an image in a frame, the program also plays a sound for each image. For every photo in a program's collection, there is a unique sound. That sound is played only when it's corresponding photo is displayed.

3 The Source Photos

Because American memory is the subject of this project, the photos are supplied by a Federal agency responsible for maintaining our collective memory, the National Archives and Records Administration. NARA maintains a massive collection of digitized documents on-line available at `http://search.nara.gov/`.

I collected the photos for this project by downloading a few photos through this search form, examining the URLs that point to these photos, and then writing a perl script to try every possible variation on those URLs. As a result of this process, I had over 100,000 files, most of which were scanned in typewritten documents. I then went through all these documents manually rejecting anything that was not a photo. In addition to rejecting non-photos, I also rejected the following classes of photos:

- badly damaged photos
- 19th Century portrait photos
- Presidential photos that are not visually interesting
- 19th Century landscape photos
- 19th Century stereograph photos

The NARA collection also includes several propoganda posters. I initially included these, but later decided to reject them since they weren't photos.

4 The Sounds

The sounds are synthesized using a synthesis technique where a wave function is dynamically altered by connecting control signals to various control points. The wave function has a fixed number of control points, but the position of the points can be dynamically altered. The control points are value/duration pairs where *value* is the value the wave function should have at that control point, and *duration* is the relative duration until the next control point. It's not quite the same as defining the control points on the wave function as x, y coordinates. The difference is that the durations are all relative to eachother, and all the durations added up are equal to one period of the waveform. So the value/duration pairs are dynamically translated into x, y coordinates, and then these points are connected to eachother by cosine segments to generate the final wave function.

The sounds for this project have an oscillator and an amplitude envelope that are both generated using the above technique. The signals controlling the oscillator and the envelope are simple cosine oscillators.