6.111 final project proposal:
A volumetric LED dot-matrix display

David Wyatt
Lawrence Wujanto

Project description

Many aspects of humanity's endeavours require the effective visualisation of three-dimensional data:
- studying the structures and interactions of biochemicals
- designing a new space vehicle
- extracting a relationship from multi-variable plots in the social sciences
- diagnosing a patient's illness from non-invasive scans
- planning a new sculpture
- ...or playing the latest computer game!

Traditional methods for displaying 3D data exclusively involve flat (2D) displays that somehow give the illusion of depth. These range from (albeit sophisticated) rendering techniques that ultimately generate a flat image, to displays that direct a different image to each eye (through head-mounted displays, coloured filter glasses or optical methods involving lenticular surfaces or parallax barriers). However, these all suffer a number of disadvantages, related to the fact that the images do not occupy the same space as the observer:

- Images typically have a limited angle of view and/or are only visible with special goggles
- Observer cannot interact with images in intuitive ways (e.g. pointing out a part of the image to a colleague with one's hand)
- Display can give erroneous impressions of relative size/scale

These shortcomings could be overcome by a true volumetric display – one where the image of a volume of space actually occupies a volume (composed of “voxels”, volumetric pixels). A number of methods for constructing such displays have been attempted in the past, including the following:
- <u>Swept volume methods</u> ([1],[2]) – projecting light onto a moving surface such that the reflected projection appears to originate from the appropriate location within the volume
- <u>Dot-matrix LED cubes</u> ([3], [4], [5], [6]) – individual LEDs in a lattice form the voxels for displaying images
- <u>Solid-state fluorescence</u> ([2])– invisible laser beams excite specific regions within a photonic crystal to emit visible light

It is proposed to build a volumetric display that uses a cubic lattice of LEDs – with, as a compromise between resolution and complexity, 512 voxels – and one or more systems to generate data to be displayed on it.

Overall system architecture

The system will be divided into modules as shown in the block diagram, shown in Figure 1.

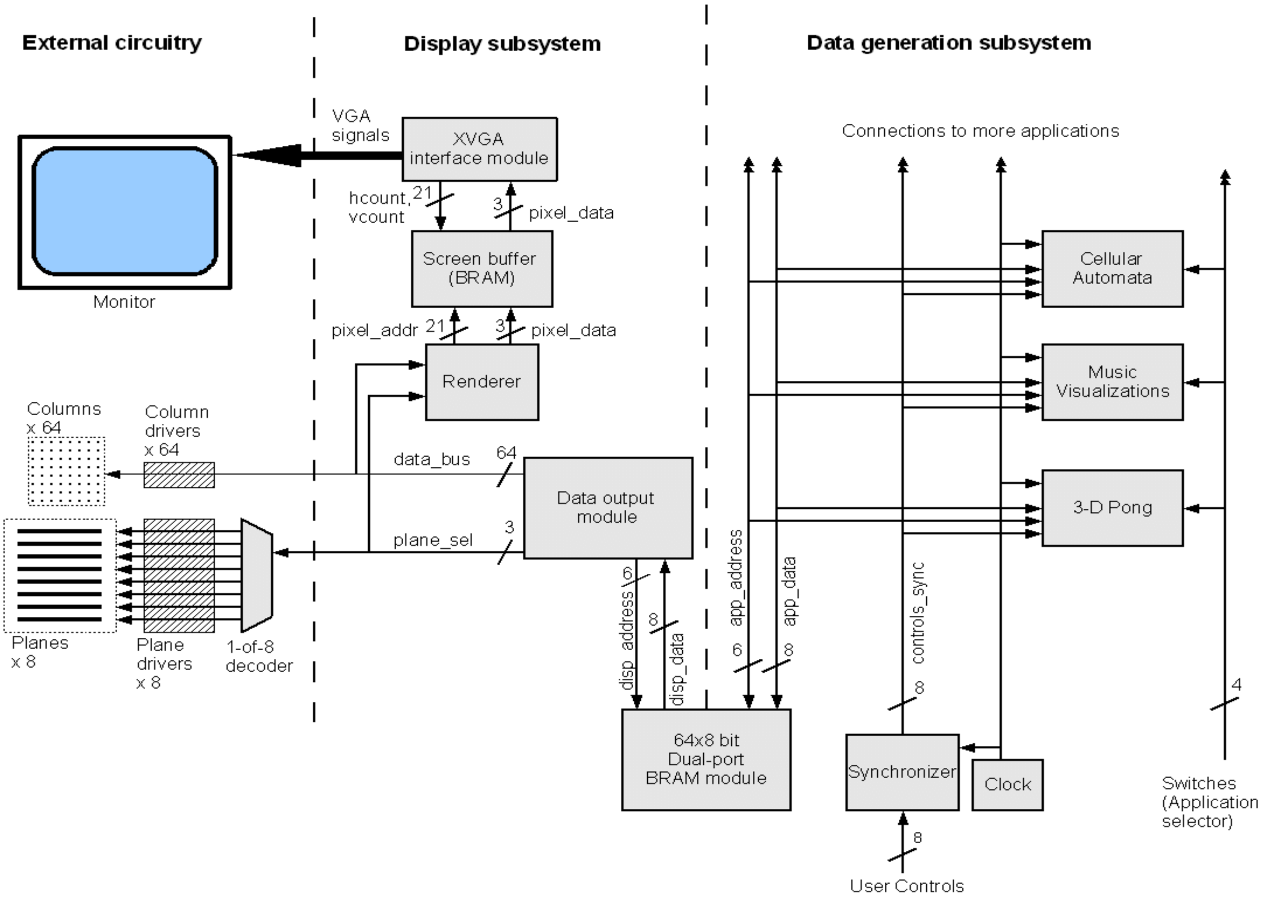

Figure 1 - System block diagram.

Data generation subsystem

The data generation subsystem consists of a collection of applications that may be displayed on the cube, and a means of selecting the relevant application to be run. The RAM module, which acts as an interface between the data generation and display subsystems, is constantly updated with the desired output from the relevant application.

Since the modular design of the data generation subsystem allows for the simple addition of new modules to the system, we shall create as many applications as the project allows time for. The following are a few examples of what we intend to implement.

3D Pong

A natural extension of the popular two-dimensional game, 3D pong is a one-player game which involves defending a surface of the cube from being struck by a bouncing "ball", by manipulating a paddle which deflects the ball. The "ball" bounces off all surfaces of the cube, with a velocity that may be manipulated by the user, in order to vary the difficulty of the game. The paddle may move in a horizontal plane at the bottom of the cube, allowing it to defend the base of the cube from the ball. To add to the complexity and challenge of the game, the direction of the ball's bounce off the paddle is a function of where the ball strikes it.

3D cellular automata

Cellular automata, a popular form of which is the Game of Life, involves a grid of cells which may each be either “live” or “dead”, and a set of rules which dictate the next state of each cell, as a function of the state of its neighbouring cells.

By using each LED to represent a cell in the system, our cube can be used to display the progression of a cellular automata system. We intend to implement cellular automata in two operation modes. The first mode uses a 2D Game of Life engine to manipulate cells in the top plane of LEDs, and then propagates past histories of the system through the lower planes of LEDs, thus allowing the time progression of the system to be seen. The second mode of the system involves extending the rules of Game of Life to three dimensions, and using the entire cube of LEDs as the playing field. This would require some trial and error to establish a new set of rules that would give the system a good balance of “live” and “dead” cells.

Music visualisations

Various characteristics of an audio signal, such as volume, pitch and rhythm, can be used to generate a graphic visualisation of sound. Although somewhat limited by the monotone colour range of the LED cube, we believe that interesting patterns can still be created on the cube by propagating information about the sound across the volume of the cube in waves.

This audio information may consist of the amplitude of the sound with a threshold function applied to it, or some representation of the frequency content of the sound.

Display of 3D data stored on a CompactFlash card

By reading stored 3D bitmaps off a CompactFlash card, we intend to allow the displaying of 3D images direct from a PC/laptop. An interface for selecting files stored on the card may be implemented using the alphanumeric display and user push buttons.

The displayed image may then be used as the initial conditions for the cellular automata module, providing a convenient way for a cellular automata system to be set up.

Display subsystem

Subsystem specification

- Transfers arbitrary data representing spatial patterns from “space buffer” RAM to 8 x 8 x 8 matrix of LEDs
- Refresh rate of display greater than 75Hz to exploit persistence of vision
- Auxiliary VGA output displays orthographic or true 3d image of display state at 1024 x 768 resolution

Voxel addressing

Owing to the fact that the number of output pins available on the labkit (192) is less than the proposed number of LEDs in the display (512), it will not be possible to address each LED individually and thus a multiplexing system must be used. The simplest way to implement this, without many auxiliary components, is to trade display update rate and LED duty cycle for decreased complexity and I/O pin requirement.

Preliminary experiments have determined that a 1 in 8 duty cycle for a high-brightness LED (3000mcd red) gives an acceptable level of light output. This suggests a multiplexing strategy where each plane of 64 LEDs is driven in sequence with the correct data, requiring 64 output pins for the LEDs plus 3 to specify the current active plane.

A simple implementation would be a passive-matrix display, in which (for example):
- All LED cathodes within a horizontal plane are wired together

- All anodes within a vertical column are wired together

With this polarity, the selected cathode plane should be driven low while all others are driven high, and simultaneously the data to be displayed on that plane is presented on the columns in uninverted logic (i.e. if the column is high the LED will be on).

Required data sizes and rates

- The size of the space buffer RAM required is 512 bits = 64 bytes. This can easily be accommodated within a single BRAM in the labkit's FPGA.
- In order for the display to be flicker-free, an update rate of at least 75Hz is required ([7]).
- This implies that the plane rate will be 8 x 75Hz = 600Hz, with each frame consisting of 64 bits.
- The eight bytes comprising the next frame's data must be fetched from the space buffer RAM, one byte at a time, during a single frame's appearance on the display.
- The byte read rate from the RAM will thus be 8 x 600Hz = 4.8KHz.
- Since only 8 bits can be read from the RAM at once, the 64 bits comprising the current plane's data must be held constant while the next plane's data is fetched – thus a two-stage pipeline will be needed, the last stage one 64-bit register and the previous stage 8 individually gated 8-bit registers.

Power requirements

The type of high-brightness LEDs that will be used in this project each draw around 30mA at 2V, and pass less than 30μA when reverse biased. Therefore, the maximum ratings for the driver circuitry will be as follows.

- Per plane:
  - when active, at maximum 64 LEDs will be on – *plane driver must sink 1.92A*
  - when inactive, at maximum 64 LEDs will be reverse biased – *plane driver must source 1.92mA*
- Per column:
  - when high, 1 LED will be on and 7 off – *column driver must source 30mA*
  - when low, 1 LED will be off and 7 reverse biased – *column driver must sink 0.21mA*

Eight 2A inverting MOSFET drivers could conveniently be used for each plane, either in combination with a row select decoder or driven directly by FPGA output pins. As the FPGA outputs can only source or sink up to 24mA they would not be suitable for direct column drivers themselves; however, TTL inverters or buffers (along with the appropriate resistors) would suffice.

Since only one plane will be active at once, the current supply requirement for the entire system will not greatly exceed 2A at under 5V (in addition to the power requirement of the FPGA and peripherals). The bench power supplies in the laboratory would therefore be sufficient.

Mechanical design and construction

For simplicity, and to reduce obscuring of the display by a support frame, the structure that holds the LEDs in the lattice will be built by soldering together the LEDs' own wires and some extra strands of similar-gauge stripped solid core wire. The proposed arrangement is shown in Figure 2 and Figure 3.

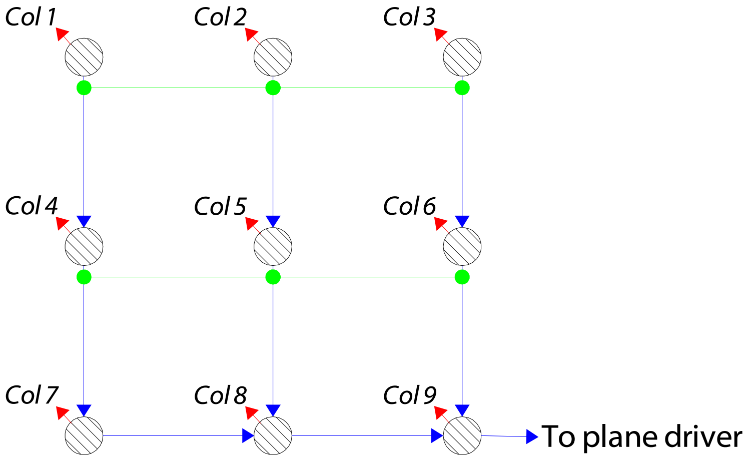

Figure 2 - Top view of wiring within a horizontal plane.

The LED cathodes (blue) are joined together in rows within each plane, with each row joined to the next at one end; supplementary wires (green) provide mechanical stability. The LED anodes (red) are connected together in columns.

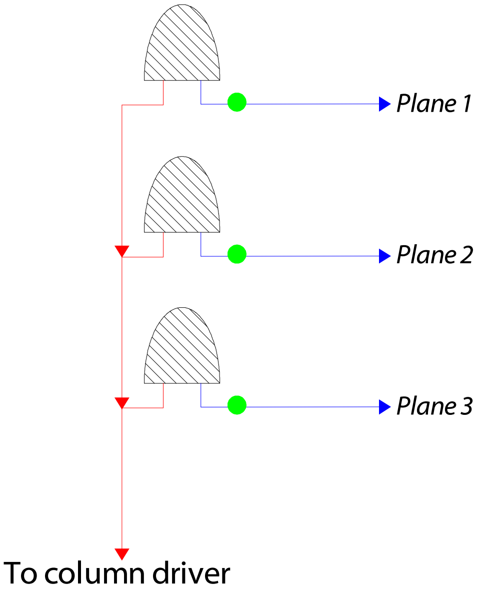

Figure 3 - Side view of wiring within a vertical column.

Such a design would allow the LEDs to be soldered together in planes of 64 (wiring up the cathodes), which are then stacked into a cube (soldering together the anodes). To assist construction of the planes, a simple jig to hold the LEDs in place while soldering could be used.

If this design proves insufficiently rigid and/or robust, external strengthening structures could be built to support the LEDs; transparent perspex rods or sheets would be able to bear significantly increased load without obscuring the LEDs excessively.

Auxiliary VGA output

If unforeseen problems make it impossible to implement this subsystem successfully, or if other circumstances require the controller to be tested without the physical display hardware, it would be possible to simulate its functionality (and thus show the output of the data generation subsystem) by providing the capability to generate an image of the LED cube on a monitor via the VGA output port. Such an auxiliary output would involve using BRAMs as a screen buffer, which could store an orthographic projection of the cube with lit LEDs represented by coloured circles.

If there is time, the basic orthographic projection could be replaced with a true 3d rendering of the cube in order to allow the user to rotate the view direction, in order to gain a better appreciation of the data displayed on the cube. This would require hardware implementation of trigonometric functions (possibly through look-up tables) and floating-point arithmetic, but should be achievable in time.

Work breakdown

The work will be divided into the two partners as follows.

- Lawrence Wujanto: data generation subsystem
- David Wyatt: display subsystem

Both partners will contribute to assembly and wiring of the display hardware.

Required resources

- 6.111 Labkit
- Windows PC and Xilinx ISE software for development (monitor can also be used for displaying auxilary output)
- 512 high-brightness non-SMT LEDs
- Baseboard
- Hardware to mount/support LEDs in a lattice: tinned wire
- Bench power supply
- LED driver circuitry – 64x TTL inverters, 64x resistors, 8x inverting 2A MOSFET drivers
- (potentially) CompactFlash card

References

[1] Actuality Systems, “Perspecta 3d display”, http://www.actuality-systems.com/ (accessed 3 November 2005)

[2] Felix 3D, “Felix 2”/”solidFELIX”, http://www.felix3d.com/ (accessed 3 November 2005)

[3] James Clar, “3d display cube – white”, http://www.jamesclar.com/product/2005/3dcubewhite/ (accessed 3 November 2005)

[4] Network Wizards, “Cubatron”, http://nw.com/nw/projects/cubatron/ (accessed 3 November 2005)

[5] Todd Holoubeck, “LED cube”, http://www.toddholoubek.com/projects/ledpage/ (accessed 3 November 2005)

[6] Chris Lomont, “LED Cube”, http://www.lomont.org/Projects/LEDCube/LEDCube.php (accessed 3 November 2005)

[7] Wikipedia, “Persistence of Vision”, http://en.wikipedia.org/wiki/Persistence_of_Vision (accessed 3 November 2005)