A Ti*k*Z rendering of the Arecibo message

Travis Stenborg

Abstract

A new tutorial on the Ti*k*Z package for vector graphics in LATEX is presented. Rendering binary-encoded data of the interstellar Arecibo message, into a digital image, was used as the learning exercise. Demonstration of data indexing, multidimensional lists, nested loops, string concatenation, variable declarations, and more, are given for budding Ti*k*Z enthusiasts. The material also has potential for adaptation to custom rendering 2D data matrices more generally.

1   Introduction

The Arecibo message is a well-known historical radio transmission, sent into space to spark extraterrestrial communication [1]. Sent from the Arecibo Telescope, the message contained pictograms of topics deemed likely to engage extraterrestrial intelligence: numbers, atoms, DNA, humanity, the Solar System and even the telescope itself [3, 15].

Modern audiences might liken the Arecibo message pictograms to pixel art. Its structure is rendered concisely here with Ti*k*Z, as a simple example of generating vector graphics from within a LATEX document. More complex Ti*k*Z examples can be found elsewhere in *TUGboat* [2, 7, 8, 10, 11, 17, 18] and beyond [4, 6, 16].

2   The Ti*k*Z package

Ti*k*Z vector graphics use the `tikz` package, available with its underlying PGF (Portable Graphics Format) system, from CTAN in `graphics/pgf` (`ctan.org/pkg/tikz`). Invocation requires merely a standard document preamble inclusion:

```
\usepackage{tikz}
```

3   Message data

The Arecibo message comprised 1679 bits (a semi-prime [9]). Arranged as 73 rows of 23 bits (both prime numbers), it encodes a raster image. A new macro `\data` was defined to emulate the message data and semantic structure. The macro is shown abbreviated, below, and in full in Figure 1.

```
% Data setup: 1679 bits.
\def \data{
{0,0,0,0,0 ... 0,0,0,0,0},   % row 1
...
{0,1,1,1,1 ... 1,1,0,0,0}}   % row 73
```

4   Configuring colors

The binary message data were rendered here in black and white. Custom color names, `pixel0` and `pixel1`,

A Ti*k*Z rendering of the Arecibo message

```
\def \data{
{0,0,0,0,0,0,1,0,1,0,1,0,1,0,0,0,0,0,0,0,0,0,0},
{0,0,1,0,1,0,0,0,0,0,1,0,1,0,0,0,0,0,0,0,1,0,0},
{1,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0,1,0,1,1,0,0,1,0},
{1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,0,1,0,0,1,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0,0,0,1,1,0,1,0,0,0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0,0,0,1,1,0,1,0,0,0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0,0,0,1,0,1,0,1,0,0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
{1,1,0,0,0,0,1,1,1,0,0,0,1,1,0,0,0,0,1,1,0,0,0,0},
{1,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,1,0,0,0,0,0,0},
{1,1,0,1,0,0,0,1,1,0,0,0,1,1,0,0,0,0,1,1,0,1,0,0},
{1,1,1,1,1,0,1,1,1,1,1,1,0,1,1,1,1,1,0,1,1,1,1,1},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
{0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
{0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1},
{1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
{1,1,0,0,0,0,1,1,0,0,0,0,1,1,1,0,0,0,1,1,0,0,0,0},
{1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0},
{1,1,0,1,0,0,0,0,1,1,0,0,0,1,1,1,0,0,0,1,1,0,1,0},
{1,1,1,1,1,0,1,1,1,1,1,1,0,1,1,1,1,1,0,1,1,1,1,1},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
{0,0,0,1,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,1,0},
{0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0},
{0,0,0,0,1,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,1},
{1,1,1,1,1,0,0,0,0,0,1,1,0,0,0,0,0,0,1,1,1,1,1},
{0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0},
{0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0},
{0,0,0,1,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,1,0,0,0},
{0,0,0,0,1,1,0,0,0,0,1,1,0,0,0,0,0,1,0,0,0,0,0},
{0,0,0,0,0,0,1,1,0,0,0,1,0,0,0,0,1,1,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,1,1,0,0,0,0,0,0,0},
{0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0},
{0,0,0,0,1,1,0,0,0,0,1,1,0,0,0,0,0,1,0,0,0,0,0,0},
{0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0},
{0,0,1,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,1,0,0,0},
{0,1,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,1,0,0},
{0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0},
{0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0},
{0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0},
{0,0,0,0,1,1,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0},
{0,0,1,0,0,0,1,1,1,0,1,0,1,1,0,0,0,0,0,0,0,0,0,0},
{0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0},
{0,0,1,0,0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0},
{0,0,1,0,0,0,0,0,1,0,1,1,0,1,0,0,0,1,0,0,1,1,0,1,1},
{0,0,0,0,0,0,1,0,0,1,0,1,1,1,0,0,0,1,0,0,1,1,1,1,1,1},
{1,0,1,1,1,0,0,0,0,0,1,1,1,0,0,0,0,0,0,1,1,0,1,1,1},
{0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,0,0,1,1,1,0,1,1},
{0,0,1,0,0,0,0,0,0,1,0,1,0,0,0,0,0,0,1,1,1,1,1,1},
{0,0,1,0,0,0,0,0,1,0,1,0,0,0,0,0,0,1,1,0,0,0,0},
{0,0,1,0,0,0,0,0,1,1,1,0,1,1,0,0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
{0,0,1,1,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0},
{0,0,1,1,1,0,1,0,1,0,0,1,0,1,0,1,0,1,0,1,0,1,0,1},
{0,0,1,1,1,0,0,0,0,0,0,0,0,1,0,1,0,1,0,1,0,1,0,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,1,0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0},
{0,0,0,0,0,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0},
{0,0,0,0,1,1,1,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0},
{0,0,0,1,1,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0},
{0,0,1,1,0,1,0,0,0,0,0,0,0,1,0,1,1,0,0,0,0,0},
{0,1,1,0,0,1,1,0,0,0,0,0,0,1,1,0,0,1,1,0,0,0},
{0,1,0,0,0,1,0,1,0,0,0,0,1,0,1,1,0,0,0,1,0,0,0},
{0,1,0,0,0,1,0,0,1,0,0,0,1,0,0,1,0,0,0,1,0,0,0},
{0,0,0,0,0,1,0,0,0,1,0,1,0,0,0,1,0,0,0,0,0,0,0},
{0,0,0,0,0,1,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0},
{0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,1,0,0,1,0,1,0,0,0,0,0,0,0,0,0,0},
{0,1,1,1,1,0,0,1,1,1,1,1,1,0,1,0,0,1,1,1,1,0,0,0}}
```

Figure 1: The Arecibo message, 1679-bit data stream T_EX macro, and 2D TikZ render.

Travis Stenborg

were defined with `\definecolor`. Mapping the name suffixes to the possible data values (0, 1), facilitates convenient rendering later.

```
% Configure colors.
\definecolor{pixel0}{rgb}{1,1,1}   % white
\definecolor{pixel1}{rgb}{0,0,0}   % black
```

5 Message rendering

TikZ graphics may be specified via *path* and supporting commands inside the `tikzpicture` environment. Here is the code, with explanations following:

```
% Render message.
\begin{tikzpicture}
\foreach \row [count=\y] in \data {
  \foreach \bit [count=\x] in \row {
    \fill[color=pixel\bit] (0.25*\x, -0.25*\y)
      rectangle +(0.25, 0.25);
  }
}
\end{tikzpicture}
```

Message data were read here bit-by-bit via the loop command `\foreach`, using the syntax `\foreach` ⟨*variable*⟩ `[`⟨*options*⟩`]` `in` ⟨*list*⟩ `{`⟨*path commands*⟩`}`. The 2D structure of the data was traversed by nesting `\foreach`, using its `count` option to index rows along the image y-axis, and bits along the x-axis.

The `\fill` command rendered each bit on a Cartesian canvas as it was read. The rendering (x, y) position was a function of the `count`-based indexing. Square pixels were specified by a `rectangle` extending from the local render position by 0.25 units along the x and y axes, i.e. `+(0.25, 0.25)`.

Finally, bit values `\bit` were used to dynamically concatenate a target color name, leveraging the suffix-value mapping implemented earlier.

The end result appears in Figure 1.

6 Discussion

This material was intended to be an educational resource for TikZ. It has the side benefit however, of demonstrating a LaTeX-friendly method for custom vector graphics rendering of 2D data matrices. Many applications might be imagined. An example is visualizing hydrologic time-series data; the Arecibo message structure has already been noted as analogous to raster hydrographs [5].

For those interested in pixel-by-pixel image composition, packages such as `pixelart` [12], `pxpic` [14] or `pixelarttikz` [13] may also be of value.

Acknowledgements

References

[1] D. Atri, J. DeMarines, J. Haqq-Misra. A protocol for messaging to extraterrestrial intelligence. *Space Policy*, 27(3):165–169, 2011.

[2] I. Borja. An introduction to automata design with TikZ's automata library. *TUGboat* 44(1):102–107, 2023. `doi.org/10.47397/tb/44-1/tb136prado-automata`

[3] V.V. Demidov. *DNA Beyond Genes: From Data Storage and Computing to Nanobots, Nanomedicine, and Nanoelectronics*, ch. Hiding and Storing Messages and Data in DNA, pp. 7–23. Springer, Cham, 2020.

[4] G. Grätzer. *More Math Into LaTeX*. Springer, Cham, 5th ed., 2016.

[5] R.B. Koehler. *Raster-based Analysis and Visualization of Hydrologic Time-Series*. Ph.D. thesis, U. of Arizona, 2004.

[6] S. Kottwitz. *LaTeX graphics with TikZ*. Packt, Birmingham, 2023.

[7] C. Maggi. The DuckBoat: The Morse code of TikZ. *TUGboat* 39(1):21–26, 2018. `tug.org/TUGboat/tb39-1/tb121duck-tikz.pdf`

[8] C. Maggi. The DuckBoat: You do not need to be Neo to cope with a TikZ matrix. *TUGboat* 41(1):20–25, 2020. `tug.org/TUGboat/tb41-1/tb127duck-matrix.pdf`

[9] M.A. Makukov, V.I. shCherbak. SETI *in vivo*: testing the we-are-them hypothesis. *Int. J. Astrobiol.*, 17(2):127–146, 2018.

[10] A. Mertz, W. Slough. Graphics with PGF and TikZ. *TUGboat* 28(1):91–99, 2007. `tug.org/TUGboat/tb28-1/tb88mertz.pdf`

[11] A. Mertz, W. Slough. A TikZ tutorial: Generating graphics in the spirit of TeX. *TUGboat* 30(2):214–226, 2009. `tug.org/TUGboat/tb30-2/tb95mertz.pdf`

[12] L. Paternault. *The pixelart package*. `ctan.org/pkg/pixelart`

[13] C. Pierquet. *The PixelArtTikz package*. `ctan.org/pkg/pixelarttikz`

[14] J.P. Spratte. *The pxpic package*. `ctan.org/pkg/pxpic`

[15] Staff at the National Astronomy, Ionosphere Center. The Arecibo message of November, 1974. *Icarus*, 26(4):462–466, Dec. 1975.

[16] M.R.C. van Dongen. *LaTeX and Friends*. Springer, Berlin, 2012.

[17] Z. Walczak. Graphics in LaTeX using TikZ. *TUGboat* 29(1):176–179, 2008. `tug.org/TUGboat/tb29-1/tb91walczak.pdf`

[18] K. Wolcott. Three-dimensional graphics with PGF/TikZ. *TUGboat* 33(1):102–113, 2012. `tug.org/TUGboat/tb33-1/tb103wolcott.pdf`

⋄ Travis Stenborg
Sydney, Australia
ORCID 0000-0002-2693-9628

A TikZ rendering of the Arecibo message