A TikZ rendering of the Arecibo message
Travis Stenborg

Abstract
A new tutorial on the TikZ 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 TikZ 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 TikZ, as a simple example of generating vector graphics from within a \LaTeX{} document. More complex TikZ examples can be found elsewhere in TUGboat [2, 7, 8, 10, 11, 17, 18] and beyond [4, 6, 16].

2 The TikZ package
TikZ vector graphics use the \texttt{tikz} package, available with its underlying PGF (Portable Graphics Format) system, from CTAN in \texttt{graphics/pgf} (ctan.org/pkg/tikz). Invocation requires merely a standard document preamble inclusion:

\begin{verbatim}
\usepackage{tikz}
\end{verbatim}

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 \texttt{\textbackslash data} was defined to emulate the message data and semantic structure. The macro is shown abbreviated, below, and in full in Figure 1.

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

4 Configuring colors
The binary message data were rendered here in black and white. Custom color names, \texttt{pixel0} and \texttt{pixel1},

\begin{verbatim}
\color{pixel0}
\end{verbatim}

A TikZ rendering of the Arecibo message

doi.org/10.47397/tb/44-3/tb138stenborg-arecibo
Figure 1: The Arecibo message, 1679-bit data stream \TeX\ macro, and 2D \LaTeX{} render.
were defined with \texttt{\textbackslash definecolor}. Mapping the name suffixes to the possible data values \((0, 1)\), facilitates convenient rendering later.

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

5 Message rendering

\textsc{Ti\kern .1667em kZ} graphics may be specified via \texttt{path} and supporting commands inside the \texttt{tikzpicture} environment. Here is the code, with explanations following:

\begin{verbatim}
% 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}
\end{verbatim}

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

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

Finally, bit values \texttt{\textbackslash 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 \textsc{Ti\kern .1667em kZ}. It has the side benefit however, of demonstrating a \textsc{B\kern .1667em T\kern .1667em E\kern .1667em X}-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 \texttt{pixelearth} [12], \texttt{pxpic} [14] or \texttt{pixelarttikz} [13] may also be of value.

Acknowledgements

This work was supported by the Australian Research Council Training Centre in Data Analytics for Resources and Environments (project ICI9010031).

References


\begin{itemize}
\item Travis Stenborg
  Sydney, Australia
  ORCID 0000-0002-2693-9628
\end{itemize}

A \textsc{Ti\kern .1667em kZ} rendering of the Arecibo message