tikz-penciline
Hand drawing with PGF/TikZ

Sébastien Gross
seb•ατ•chezwam•∂θτ•org
@renard_0
https://github.com/renard/tikz-penciline

2014/12/18 v1.0 initial version

`tikz-penciline` is based on *percusse* answer from StackExachange's *simulating hand drawn lines*[1]. The original idead is to make a `tikzpicture` look like if it was hand drawn.

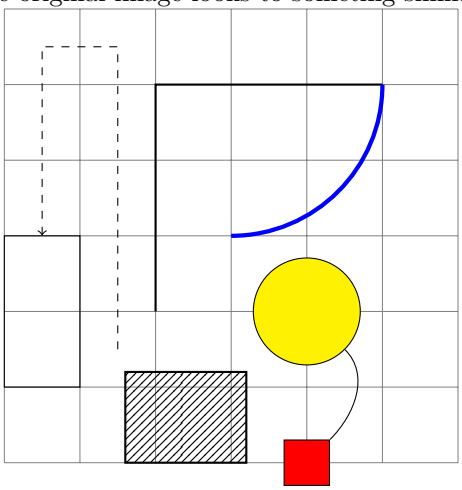
Contents

This documentation was compiled on Mac OS X using XeLaTeX 0.99991, PGF 3.0.0, `tcolorbox` 3.21 on December 29, 2014.

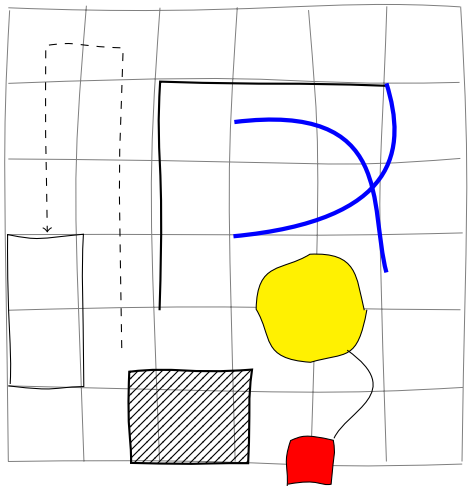
[1] http://tex.stackexchange.com/questions/39296

1 Overview

The original image looks to someting similar to:



The **penciline** version looks to someting similar to:



```
\begin{tikzpicture}[penciline={jag ratio=1}]
  \draw[decorate,penciline={jag ratio=2},style=help lines] (-2,-2)
  grid[step=1cm] (4,4);
  \draw[decorate,thick] (0,0) -- (0,3) -- (3,3);
   %% This is supposed to be an arc!!
  \draw[decorate,penciline={jag ratio=0,arc angle base2=90},ultra thick,blue] (3,3) arc (0:-90:2cm);
  \draw[decorate,penciline={jag ratio=0},ultra thick,blue] (3,0.5) arc (0:90:2cm);
  \draw[decorate,thick,pattern=north east lines] (-0.4cm,-0.8cm) rectangle (1.2,-2);
   %% That's not even an ellipse !!
  \node[decorate,penciline={jag ratio=0,arc radius base1=0.4,arc radius base2=.6},
    draw,inner sep=0.5cm,fill=yellow,circle] (a) at (2,0) {};
  \node[decorate,draw,inner sep=0.3cm,fill=red] (b) at (2,-2) {};
   %% This was supposed to be an edge!!
  \draw[decorate,penciline={jag ratio=0}] (b) to[in=-45,out=45] (a);
  \node[decorate,draw,minimum height=2cm,minimum width=1cm] (c) at (-1.5,0) {};
  \draw[decorate,->,dashed] (-0.5cm,-0.5cm) -- (-0.5cm,3.5cm)  -| (c.north);
\end{tikzpicture}
```

2 Installation

To install the **tkiz-penciline** package copy its directory to either to

- `$TEXHOME/tex/latex/`
- `$TEXMFHOME/tex/latex/`
- `~/texmf/tex/latex/`
- `~/Library/texmf/tex/latex/`

3 Usage

Basically you only have to declare **penciline** and **decorate** option to path you want to look hand drawn.

3.1 customization

jag ration=⟨*float*⟩ (2.0)

This value controls the *jag* value. The higher the it is, the most deformed the drawing will be. Best value are between 2 and 5.
If the path is a circle or an arc you should set **jag ratio** to 0 for better results.

segment x base1=⟨*float*⟩ (0.5)

This is the base for the x coordinate of first control point. The formula is given by:
length * (**segment x base1** + **segment x ratio1** * rnd)
This means by default that that the x coordinate is about in the middle of the segment.

segment x ratio1=⟨*float*⟩ (0.25)

This is the randomized ratio part of the x coordinate of first control point (see `segment x base1`).

segment x base2=⟨*float*⟩ (0.5)

Base for second control point (see `segment x base1`).

segment x ratio2=⟨*float*⟩ (0.25)

Random ratio for second control point (see `segment x ratio1`).

arc angle base1=⟨*float*⟩ (20)

This is the base of the angle of the polar coordinate for the first control point. The formula is given by:
arc angle base1 + rand * **arc angle ratio1**
This means that the angle is **arc angle base1** ± **arc angle ratio1**.

`arc angle ratio1`=⟨*float*⟩ (10)

This is the ratio (variable) part of the angle of the polar coordinate for the first control point (see `arc angle base1`).

`arc radius base1`=⟨*float*⟩ (0.3)

This is the base of the radius of the polar coordinate for the first control point. The formula is given by:
length * `arc radius base1` + rand * `arc radius ratio1`

`arc radius ratio1`=⟨*float*⟩ (0.05)

This is the ratio (variable) part of the radius of the polar coordinate for the first control point (see `arc radius base1`).

`arc angle base2`=⟨*float*⟩ (-90)

Base angle for the second control point. See `arc angle base1` → P. 3.

`arc angle ratio2`=⟨*float*⟩ (-10)

Variable part of the angle for the second control point. See `arc angle ratio1`.

`arc radius base2`=⟨*float*⟩ (0.7)

Base of the radius for the second control point. See `arc radius base1`.

`arc radius ratio2`=⟨*float*⟩ (0.05)

Variable part of the radius for the second control point. See `arc radius ratio1`.

`x offset`=⟨*float*⟩ (1)

Maximum x offset for target point.

`y offset`=⟨*float*⟩ (1)

Maximum y offset for target point.