Coffin Usage

Eureka

May 31, 2024

Contents

1 introduce

Current interfaces of `xcoffins` package is stable, See more at TEX SE[1]

1.1 What is Coffin

coffin is a box not only containing “something you type”, but also includes some information about it, like “size, shape”.

Then we can **align 2 or more coffins**. How doese this archieve ? “coffins” privides a series of ‘poles(lines)’ to connect some typical points on a coffin. We call these special points a “handles”. Then 2 coffins can be align be describing the relationship between a handle on one confin with a handle a handle on the second.

1.2 handles

Handle positions are different in horizontal mode and vertical mode, next we consider coffin in horizontal mode. The following special handles are pre-defined:

- (hc, vc): center of a conffin
- (hc, t): horizontal center and top
- ⋯

we can simply remeber these handles by: **(right=r, horizontal center=hc, right=r)**, and **(top=t, vertical center=vc, bottom=b)**. You can see `(hc, vc)` contains 2 lines `line-1=hc, line-2=vc`, then this point is the intersection.

Aside from pre-defined handles, each time you add a poles, (maybe) there will be some new handles added to this coffin. That’s “handles of a coffin is dynamic”.

[1] https://tex.stackexchange.com/a/397835/294585

2 create and use coffins

let's begin with a simple example: before align any 2 coffins, you must:

- create a coffin (not local)
- add contents to this coffin

All coffin operations are local to current TEX group. Decaler a coffin using `\NewCoffin <\mycoffin >`, `\mycoffin` is the name of this new coffin.

Then we can add content to `\mycoffin` , the content add to this coffin can be typset in 2 modes

- `\SetHorizontalCoffin \mycoffin {<your content>}`(**you can't line-break in this coffin(box)**)
- `\SetVerticalCoffin \mycoffin {<width>}{<your content>}`

the the standard poles are set up based on the size of your content. Then you can use `\mycoffin` using command `\TypesetCoffin \mycoffin` or `\usebox {\mycoffin }`(Legacy in TEX).

i add something in this coffin, and this coffin is in horizontal mode.

i add something in this coffin, and this coffin is in vertical mode.

2.1 coffin poles

some standard poles list below:

- **l**: left-hand edge
- ···
- **H**: a pole running along the **baseline** of the typeset material contained in the coffin

some extra poles in vertical mode material:

- **B**: a pole running along the **baseline** of the material at the **bottom** of the coffin
- **T**: a pole running along the **baseline** of the material at the **top** of the coffin.

2.2 set poles

You can add some horizontal or vertical line a exist coffin, use the command (take `\mycoffin` for example):

- `\SetHorizontalPole \mycoffin {<pole>}{<offset>}`. `<pole>` will be located at the `<offset>` from the **baseline** of the coffin.

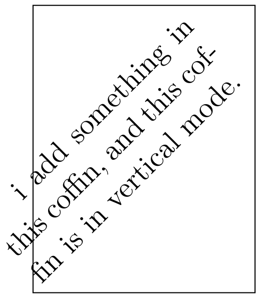
- `\SetVerticalPole \mycoffin {<pole>}{<offset>}`. `<pole>` will be located at the `<offset>` from the **left-hand** edge of the bounding box of the coffin.

You should notice that

Total height of coffin(base to top) = `\TotalHeight` = `\Height` + `\Depth`

2.3 rotate coffin

Use command `\RotateCoffin <coffin>{<angle>}` to ratate **counter-clockwise** about a **referance point**. This process will rotate both the coffin content and poles. Use the above `\myvcoffin` for example, by command `\RotateCoffin \myvcoffin {45}` will result in:



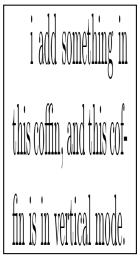
Multiple rotations will not result in the bounding box of the coffin growing, it is just the bounding box become slope.

2.4 scale coffin

2.4.1 absolute values

If you are not satisfied with your coffin, you can resize it(scale to a fixed dimension in width or length), redefine its `<width>, <height>`. For that the coffin `\mycoffin` can't line-break, so the `<height>` is fixed. I'll use `\myvcoffin` . command `\ResizeCoffin \myvcoffin {4em}{2em+6em}` will result in:

i add something in
this coffin, and this cof-
fin is in vertical mode.



2.4.2 scale factors

To scale both length and with by given real numbers(factors) in x and y directions, use command `\ScaleCoffin <coffin>{<x-scale>}{<y-scale>}`. command `\ScaleCoffin`

`\myvcoffin {2}{1}` will result in:

i add something in this coffin, and this coffin is in vertical mode.	i add something in this coffin, and this coffin is in vertical mode.

`\ResizeCoffin` and `\ScaleCoffin` can be used interchangeably, depend upon the context.

2.5 join coffin

Join coffin is one of the most important in operations of coffins. `xcoffins` provides 2 command for joining coffins: `\JoinCoffins` , `\JoinCoffins *`. The difference between these two is that: the first one will expand the bounding box of "parent coffin" while the "child coffin" will protrude outside of the bounding box of the updated "parent coffin".

How do these 2 coffins align (attacked)?

- First you select 2 `<handles>` in each of coffins, decaler these `<handles>` by intersection of 2 lines, that's `[<coffin1-pole1>, <coffin1-pole2>]`.
- Second, after the 2 `<handles>` are setted up, align the 2 coffins by assingn the `(x-offset>, <y-offset>)` to align these 2 coffins.

see figure (1).

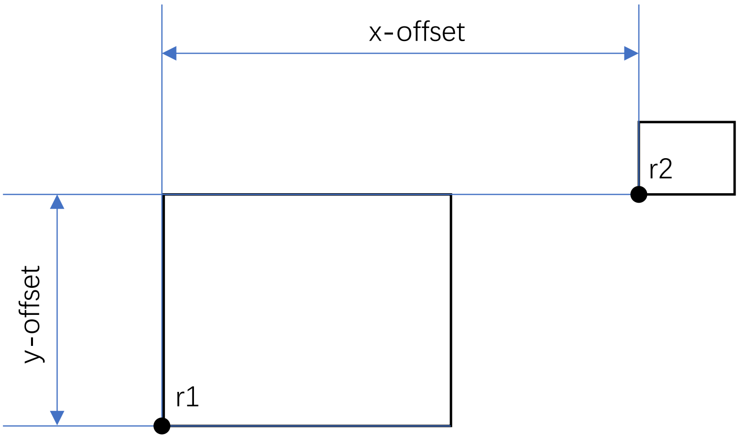


Figure 1: how to join coffins

So the full command to align 2 coffins is

```
\JoinCoffins
    <coffin1> [<coffin1-pole1>, <coffin1-pole2>]
    <coffin2> [<coffin2-pole1>, <coffin2-pole2>]
    (<x-offset>, <y-offset>)
```

Notice that "This `<offset>` is an optional argument, and if it is not given then `(0pt, 0pt)` is used." When `\JoinCoffins` is used the new bounding box is the smallest rectangle containing the bounding boxes of the two input coffins. This may introduce addtional white space in this new "bounding box" when you rotate the joined (two) coffin(unless one coffin entirely overlaps the other).

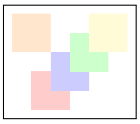
Like applying "RortateCoffin" to a single coffin, Rotation of coffins will take account of the extent of the material after rotation when re-calculating the bounding box. This means that **no unnecessary white space** will be added on rotation as well.

The poles of the two input coffins are preserved within the structure of the updated coffin, consisting the original "poles" in **both** coffins.

A simple joining operations as follows, the full code is:

```
\def\bb#1{\color{#1!20!white}\rule{0.2 in}{0.2 in}}
% create coffins
\NewCoffin\OutputCoffin
\NewCoffin\RedCoffin
\NewCoffin\BlueCoffin
\NewCoffin\GreenCoffin
\NewCoffin\YellowCoffin
\NewCoffin\OrangeCoffin

% set coffins
\SetHorizontalCoffin\OutputCoffin{}
\SetHorizontalCoffin\RedCoffin{\bb{red}}
\SetHorizontalCoffin\BlueCoffin{\bb{blue}}
\SetHorizontalCoffin\GreenCoffin{\bb{green}}
\SetHorizontalCoffin\YellowCoffin{\bb{yellow}}
\SetHorizontalCoffin\OrangeCoffin{\bb{orange}}

% set coffins poles
\SetHorizontalPole\RedCoffin{red-vc}{\Height/2}
\SetVerticalPole\RedCoffin{red-hc}{\Width/2}
\SetHorizontalPole\BlueCoffin{blue-vc}{\Height/2}
\SetVerticalPole\BlueCoffin{blue-hc}{\Width/2}
\SetHorizontalPole\GreenCoffin{green-vc}{\Height/2}
\SetVerticalPole\GreenCoffin{green-hc}{\Width/2}
\SetHorizontalPole\YellowCoffin{yellow-vc}{\Height/2}
\SetVerticalPole\YellowCoffin{yellow-hc}{\Width/2}
\SetHorizontalPole\BlueCoffin{blue-t}{\Height}
\SetVerticalPole\BlueCoffin{blue-r}{0pt}

% join coffin to '\OutputCoffin'
\JoinCoffins\OutputCoffin[vc, hc]\RedCoffin[vc, hc]
\JoinCoffins\OutputCoffin[red-vc, red-hc]\BlueCoffin[l, b]
\JoinCoffins\OutputCoffin[blue-vc, blue-hc]\GreenCoffin[l, b]
\JoinCoffins\OutputCoffin[green-vc, green-hc]\YellowCoffin[l, b]
\JoinCoffins\OutputCoffin[blue-t, blue-r]\OrangeCoffin[r, b]

% typset the \OutputCoffin
\usecoffin{\OutputCoffin}
```

This code will result in the following graph:



2.6 Typset Coffin

How to typset a coffin we have decalered before, use command `\TypsetCoffin` . Though we have use this command before, but we haven't introduce the full syntax of this command. The full syntax is:

```
\TypesetCoffin
    <coffin> [<pole1>, <pole2>]
    (<x-offset>, <y-offset>)
```

we can typset a coffin such that the relation between the **current reference point in docuemnt** and the `<handle>`(defined by these 2 lines) is described by the `<x-offset>` and `<y-offset>`. Typesetting a coffin is therefore analogous to carrying out an alignment where the "parent" coffin is the current insertion point.

Finally, i'll introduce some functions measuring coffins, which is useful in `<dimension expression>`.

3 Measuring Coffins

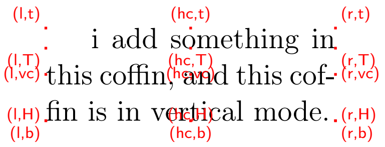
`xcoffins` provide some function to measure the `depth, length, width` of a coffin. See below:

- `\CoffinDepth <coffin>`: the depth of the coffin
- `\CoffinHeight <coffin>`: the height of the coffin
- `\CoffinTotalHeight <coffin>`: the total height of the coffin
- `\CoffinWidth <coffin>`: the width of the coffin

4 Diagnostic

Diagnostic data for following the coffin-building process is available both graphically and at the terminal.

Firstly, how to print a coffin including all the 'poles' and 'handles' in it? Use command `\DisplayCoffinHandles <coffin>{<color>}`. This command will print the coffin with all the poles and handles **graphically** at the current location in the source:



It's also possible to show a specific `<handle>` by 2 poles you defiend before. Then introduce some Diagnostic functions in terminal, `\ShowCoffinStructure` . Using this function, you may see something like the below, command `\ShowCoffinStructure \myvcoffin` :

```
Size of coffin \myvcoffin:
> ht = 34.80417pt
> dp = 0.0pt
> wd = 109.50027pt
Poles of coffin \myvcoffin:
>  l  =>  {0.0pt}{0.0pt}{0.0pt}{1000.0pt}
>  hc  =>  {54.75012pt}{0.0pt}{0.0pt}{1000.0pt}
>  r  =>  {109.50027pt}{0.0pt}{0.0pt}{1000.0pt}
>  b  =>  {0.0pt}{0.0pt}{1000.0pt}{0.0pt}
>  vc  =>  {0.0pt}{17.40208pt}{1000.0pt}{0.0pt}
>  t  =>  {0.0pt}{34.80417pt}{1000.0pt}{0.0pt}
>  B  =>  {0.0pt}{0.0pt}{1000.0pt}{0.0pt}
>  H  =>  {0.0pt}{0.0pt}{1000.0pt}{0.0pt}
>  T  =>  {0.0pt}{27.20001pt}{1000.0pt}{0.0pt}.
<recently read> }
```

Notice that the poles of a coffin are defined by four values: the x and y coordinates of a point that the pole passes through and the x- and y-components of a vector denoting the direction of the pole. (**point** + **direction** for short).